

# Multi-Party Protocols, Information Complexity and Privacy

Iordanis Kerenidis<sup>\*1</sup>, Adi Rosén<sup>†1</sup>, and Florent Urrutia<sup>‡2</sup>

<sup>1</sup>CNRS and Université Paris Diderot

<sup>2</sup>Université Paris Diderot

June 23, 2016

## Abstract

We introduce the new measure of *Public Information Complexity* (PIC), as a tool for the study of multi-party computation protocols, and of quantities such as their communication complexity, or the amount of randomness they require in the context of information-theoretic private computations. We are able to use this measure directly in the natural asynchronous message-passing *peer-to-peer* model and show a number of interesting properties and applications of our new notion: the Public Information Complexity is a lower bound on the Communication Complexity and an upper bound on the Information Complexity; the difference between the Public Information Complexity and the Information Complexity provides a lower bound on the amount of randomness used in a protocol; any communication protocol can be compressed to its Public Information Cost; an explicit calculation of the zero-error Public Information Complexity of the  $k$ -party,  $n$ -bit Parity function, where a player outputs the bit-wise parity of the inputs. The latter result establishes that the amount of randomness needed for a private protocol that computes this function is  $\Omega(n)$ .

## 1 Introduction

Communication complexity, originally introduced by Yao [Yao79], is a prolific field of research in theoretical computer science that yielded many important results in various fields. Informally, it answers the question “How many bits must the players transmit to solve a distributed problem ?” The study of the two-party case has produced a large number of interesting and important results, upper and lower bounds, with many applications in other areas in theoretical computer science such as circuit complexity, data structures, streaming algorithms and distributed computation (see, e.g., [KN97, MNSW95, GG10, SHK<sup>+</sup>10, FHW12]).

---

<sup>\*</sup>Research supported in part by ERC QCC and ANR grant RDAM.

<sup>†</sup>Research supported in part by ANR grant RDAM.

<sup>‡</sup>Research supported in part by ERC QCC and ANR grant RDAM.

A powerful tool recently introduced for the study of two-party communication protocols is the measure of *Information Complexity* (or *cost*). This measure, originally defined in [BCKO93] and [CSWY01], extends the notions of information theory, originally introduced by Shannon [Sha48], to interactive settings. It quantifies, roughly speaking, the amount of information about their respective inputs that Alice and Bob must leak to each other in order to compute a given function  $f$  of their inputs. Information complexity (IC) has been used in a long series of papers to prove lower bounds on communication complexity and other properties of (two-party) communication protocols (e.g., [BYJKS02, BBCR10, BR11, Bra12]). An interesting property of information complexity is that it satisfies a direct sum. The *direct sum* question, one of the most interesting questions in complexity theory, asks whether solving  $n$  independent copies of the same problem must cost (in a given measure)  $n$  times the cost of solving a single instance. In the case of communication complexity, this question has been studied in, e.g., [FKNN95, CSWY01, Sha03, JRS03, HJMR10, BBCR10, Kla10, Jai15] and in many cases it remains open whether a direct sum property holds.

Another important question in communication complexity is the relation between the information complexity of a function and its communication complexity. We would like to know if it is possible to compute a function by sending a number of bits which is not (too much) more than the information the protocol actually has to reveal. Put differently, is it always possible to *compress* the communication cost of a protocol to its information cost? For the two-party case it is known that perfect compression is not possible [GKR15a, GKR15b]. Still, several interesting compression results are known. The equality between information cost and amortized communication cost is shown in [BR11, Bra12], and other compression techniques are given in [BBCR10, BMY15, BBK<sup>+</sup>13, Pan15]. It remains open if one can compress interactive communication up to some small loss (for example logarithmic in the size of the input).

When trying to study the *multi-party* (i.e., where at least 3 players are involved) communication setting using similar information-theoretic methods, such as IC, one encounters a serious problem. The celebrated results on information-theoretic private computation [BOGW88, CCD88] state that if the number of players is at least 3, then *any function* can be computed by a randomized protocol such that *no information* about the inputs is revealed to the other players (other than what is implied by the value of the function and their own input). Thus, in the multi-party case, the IC of any function  $f$  is 0 (or only the entropy of  $f$ , depending on the definition of IC), and cannot serve to study multi-party protocols.

For this reason, information complexity has rarely been used in the multi-party setting, where most results have been obtained via combinatorial techniques. Among the interesting works on multi-party setting are [PVZ12, WZ14] which introduce the techniques of *symmetrization* and *composition*, and [CRR14, CR15] which study the influence of the topology of the network. One notable exception is the work of Braverman et al. [BEO<sup>+</sup>13] which studies the *set-disjointness* problem using information theoretic tools. Braverman et al. provide almost tight bounds in the so-called coordinator model (that differs from the more natural peer-to-peer model) by analyzing the information leaked between the players but also the information obtained by the coordinator itself. The set disjointness problem is maybe one of the most extensively studied problem in communication complexity (cf. [Bra12, BYJKS02, CKS03, Gro09, Jay09,

BGPW13]). This line of research was followed by [CM15] which also uses information complexity to obtain tight bounds on the communication complexity of the function *Tribes* in the coordinator model. Information complexity is also used in [BO15] to study set-disjointness in the broadcast model.

A number of sub-models have been considered in the literature for the multi-party computation protocols setting: the *number in hand model* (NIH), where each player has a private input, is maybe the most natural one, while in the *number on the forehead model* (NOF), each player  $i$  knows all inputs  $x_j$ ,  $j \neq i$ , i.e., the “inputs” of all players except its own. As to the communication pattern, a number of variants exist as well: in the *blackboard* model, the players communicate by broadcasting messages (or writing them on a “blackboard”); in the *message passing* model, each pair of players is given a private channel to mutually communicate (for more details on different variants see [KN97]). Most of the results obtained in multi-party communication complexity were obtained for the NOF model and/or the blackboard model. The present paper studies, however, the NIH, message passing (peer to peer) model, which is also the most closely related to the work done on message passing protocols in the distributed computing and networking communities.

## 1.1 Our contributions

Our main goal is to introduce novel information-theoretical measures for the study of number in hand, message-passing multi-party protocols, coupled with a natural model that, among other things, allows private protocols (which is not the case for, e.g., the coordinator model)

We define the new measure of *Public Information Complexity* (PIC), as a tool for the study of multi-party computation protocols, and of quantities such as their communication complexity, or the amount of randomness they require in the context of information-theoretic private computations. Intuitively, our new measure captures a combination of the amount of information about the inputs that the players leak to other players, and the amount of randomness that the protocol uses. By proving lower bounds on PIC for a given multi-party function  $f$ , we are able to give lower bounds on the communication complexity of  $f$  and on the amount of randomness needed to privately compute  $f$ . The crucial point is that the PIC of functions, in our multi-party model, is not always 0, unlike their IC.

Our new measure works in a model which is a slight restriction of the most general asynchronous model, where, for a given player at a given time, the set of players from which that player waits for a message can be determined by that player’s own local view. This allows us to have the property that for any protocol, the information which is leaked during the execution of the protocol is at most the communication cost of the protocol. Note that in the multi-party case, the information cost of a protocol may be higher than its communication cost, because the identity of the player from which one receives a message might carry some information. This is another issue when trying to use IC in the peer-to-peer multi-party setting. We are able to define our measure and use it directly in a natural asynchronous *peer-to-peer* model (and not, e.g., in the coordinator model used in most works studying the multi-party case, c.f. [DF89]). The latter point is particularly important when one is interested in privacy, since our model allows for private protocols, while this is not necessarily the case

for other models, including the coordinator model. Furthermore, if one seeks to accurately understand the natural peer-to-peer model, suppressing polylog-factor inaccuracies, one has to study directly the peer-to-peer model (see the comparison of models in subsection 3.1).

We go on to show a number of interesting properties and applications of our new notion:

- The Public Information Complexity is a lower bound on the Communication Complexity and an upper bound on the Information Complexity. In fact, it can be strictly larger than the Information Complexity.
- The difference between the Public Information Complexity and the Information Complexity provides a lower bound on the amount of randomness used in a protocol.
- We compress any communication protocol to their PIC (up to logarithmic factors), by extending to the multi-party setting the work of Brody et al. [BBK<sup>+</sup>13] and Pankratov [Pan15].
- We show that one can approach the central question of direct sum in communication complexity by trying to prove a direct sum result for PIC. Indeed, we show that a direct sum property for PIC implies a certain direct sum property for communication complexity.
- We explicitly calculate the zero-error *Public Information Complexity* of the  $k$ -party,  $n$ -bit Parity function (Par), where a player outputs the bit-wise parity of the inputs. We show that the PIC of this function is  $n(k-1)$ . This result is tight and it also establishes that the amount of randomness needed for a private protocol that computes this function is  $\Omega(n)$ . While this sounds a reasonable assertion no previous proof for such claim existed.

## 1.2 Organization

The paper is organized as follows. In section 2 we review several notations and information theory basics. In Section 3 we define the communication model and a number of traditional complexity measures. In Section 4 we define the new measure PIC that we introduce in the present paper, and in Section 5 we discuss its relation to randomness and multi-party private computation. In section 6 we discuss the existence of a direct sum property for PIC, and in Section 7 we give tight bounds for Par using PIC. We conclude the paper in Section 8.

## 2 Preliminaries

We start by defining a number of notations. We denote by  $k$  the number of players. We often use  $n$  to denote the size (in bits) of the input to each player. Calligraphic letters will be used to denote sets. Upper case letters will be used to denote random variables, and given two random variables  $A$  and  $B$ , we will denote by  $AB$  the joint random variable  $(A, B)$ . Given a string (of bits)  $s$ ,  $|s|$  denotes the length of  $s$ . Using parentheses we denote an ordered set (family) of items, e.g.,  $(Y_i)$ . Given a family  $(Y_i)$ ,  $Y_{-i}$  denotes the sub-family which is the family  $(Y_i)$  *without* the element  $Y_i$ . The letter  $X$  will usually denote the input

to the players, and we thus use the shortened notation  $X$  for  $(X_i)$ , i.e. the input to all players.  $\pi$  will be used to denote a protocol. We use the term *entropy* to talk about binary entropy.

We give a reminder on basic information theory, as introduced in [Sha48].

**Definition 2.1.** *The entropy of a (discrete) random variable  $X$  is*

$$H(X) = \sum_x \Pr[X = x] \log \left( \frac{1}{\Pr[X = x]} \right).$$

*The conditional entropy  $H(X | Y)$  is defined as  $\mathbb{E}_y[H(X | Y = y)]$ .*

**Proposition 2.2.** *For all finite set  $\mathcal{X} \subseteq \{0, 1\}^*$  and all random variable  $X$  with support  $\text{supp}(X) \subseteq \mathcal{X}$ , it holds*

$$H(X) \leq \log(|\mathcal{X}|).$$

*Moreover, if the set  $\mathcal{X}$  is prefix-free, it holds  $H(X) \leq \mathbb{E}[|X|]$ .*

**Definition 2.3.** *The mutual information between two random variables  $X, Y$  is*

$$I(X, Y) = H(X) - H(X | Y).$$

*The conditional mutual information  $I(X; Y | Z)$  is  $H(X | Z) - H(X | YZ)$ .*

The mutual information measures the change in the entropy of  $X$  when one learns the value of  $Y$ . It is non negative, and is symmetric:  $I(X, Y) = I(Y, X)$ .

**Proposition 2.4.** *For any random variables  $X, Y$  and  $Z$ ,  $I(X; Y | Z) = 0$  if and only if  $X$  and  $Y$  are independent for each possible values of  $Z$ .*

**Proposition 2.5.** *For any random variables  $X$  and  $Y$ ,  $H(X | Y) \leq H(X)$ .*

**Proposition 2.6** (Chain Rule). *Let  $A, B, C, D$  be four random variables. Then*

$$I(AB; C | D) = I(A; C | D) + I(B; C | DA).$$

**Lemma 2.7** (Data processing inequality). *For any  $X, Y, Z$ , and any function  $f$ , it holds:  $I(X; f(Y) | Z) \leq I(X; Y | Z)$*

*Proof.*

$$\begin{aligned} I(X; f(Y) | Z) &\leq I(X; f(Y) | Z) + I(X; Y | f(Y)Z) \\ &\leq I(X; Yf(Y) | Z) \\ &\leq I(X; Y | Z) + I(X; f(Y) | YZ) \\ &\leq I(X; Y | Z). \end{aligned}$$

┘

**Proposition 2.8** ([Bra12]). *Let  $A, B, C, D$  be four random variables such that  $I(B; D | AC) = 0$ . Then*

$$I(A; B | C) \geq I(A; B | CD).$$

**Proposition 2.9** ([Bra12]). *Let  $A, B, C, D$  be four random variables such that  $I(B; D | C) = 0$ . Then*

$$I(A; B | C) \leq I(A; B | CD).$$

### 3 The model

We now define a natural communication model which is a slight restriction of the most general asynchronous peer-to-peer model. Its restriction is that for a given player at a given time, the set of players from which that player waits for a message can be determined by that player's own local view. This allows us to define information theoretical tools that pertain to the transcripts of the protocols, and at the same time to use these tools as lower bounds for communication complexity. This restriction however does not exclude the existence of private protocols, as other special cases of the general asynchronous model do. We observe that without such restriction the information revealed by the execution of a protocol might be higher than the number of bits transmitted and that, on the other hand, practically all multi-party protocols in the literature are implicitly defined in our model. We also compare our model to the general one and to other restricted ones and explain the usefulness and logic of our specific model.

#### 3.1 Definition of the model

We work in the *multi-party number in hand peer-to-peer* model. Each player has unbounded local computation power and, in addition to its input  $X_i$ , has access to a source of private randomness  $R_i$ . We will use the notation  $R$  for  $(R_i)$ , *i.e.*, the private randomness of all players. A source of public randomness  $R^p$  is also available to all players. The system consists of  $k$  players and a family of  $k$  functions  $f = (f_i)_{i \in [1, k]}$ , with  $\forall i \in [1, k], f_i : \prod_{l=1}^k \mathcal{X}_l \rightarrow \mathcal{Y}_i$ , where  $\mathcal{X}_l$  denotes the set of possible inputs of player  $l$ , and  $\mathcal{Y}_i$  denotes the set of possible outputs of player  $i$ . The players are given some input  $x = (x_i) \in \prod_{i=1}^k \mathcal{X}_i$ , and for every  $i$ , player  $i$  has to compute  $f_i(x)$ . Each player has a special write-only output tape.

We define the communication model as follows, which is the asynchronous setting, with some restrictions. To make the discussion simpler we assume a global time which is *unknown* to the players. Every pair of players is connected by a bidirectional communication link that allows them to send messages in both directions. There is no bound on the delivery time of a message, but every message is delivered in finite time, and the communication link maintains FIFO order in each of the two directions. Given a specific time we define the *view* of player  $i$ , denoted  $D_i$ , as the input of that player,  $X_i$ , its private randomness,  $R_i$ , and the messages received so far by player  $i$ . The protocol of each player  $i$  runs in *local* rounds. In each round, player  $i$  sends messages to some subset of the other players. The identity of these players, as well as the content of these messages, depend on the current view of player  $i$ . The player also decides whether to write a (nonempty) string on its output tape. Then, the player waits for messages from a certain subset of the other players, where this subset is also determined by the current view of the player. Then the (local) round of player  $i$  terminates<sup>1</sup>. To make it possible for the player to identify the arrival of the

---

<sup>1</sup>The fact that the receiving of the incoming messages comes as the last step of the (local) round comes only to emphasize that the sending of the messages and the writing on the output tape are a function of only the messages received in previous (local) rounds.

*complete* message that it waits for, we require that each message sent by a player in the protocol is self-delimiting.

Denote by  $D_i^j$  the view of player  $i$  at the end of local round  $j$ ,  $j \geq 0$ , where the beginning of the protocol is considered round 0. Formally, a protocol  $\pi$  is defined by a sequence of functions for each player  $i$ , parametrized by the *local* round  $j$ ,  $j \geq 1$ :

- $\overline{S}_i^j : D_i^{j-1} \rightarrow 2^{\{1, \dots, k\} \setminus \{i\}}$ , defining the set of players to which player  $i$  sends the messages.
- $m_{i,q}^j : D_i^{j-1} \rightarrow \{0, 1\}^*$ , for all  $q \in \overline{S}_i^j(D_i^{j-1})$ , defining the content of the messages player  $i$  sends. Each such message has to be self-delimiting.
- $O_i^j : D_i^{j-1} \rightarrow \{0, 1\}^*$ , defining what the player writes on the output tape. Each player can write on its output tape a non-empty string only once.<sup>2</sup>
- $S_i^j : D_i^{j-1} \rightarrow 2^{\{1, \dots, k\} \setminus \{i\}}$ , defining the set of players from which player  $i$  waits for a message.

We note that the model does not impose “coherence” between the players. That is, the model does not preclude the possibility that a certain player waits indefinitely for a message that is never sent to it.

We define the transcript of the protocol of player  $i$ , denoted  $\Pi_i$ , as the concatenation of the messages read by player  $i$  from the links of the sets  $S_i^1, S_i^2, \dots$ , ordered by local round number, and within each round by, say, the index of the player. We denote by  $\overleftrightarrow{\Pi}_i$  the concatenation of  $\Pi_i$  together with a similar concatenation of the messages sent by player  $i$  to the sets  $\overline{S}_i^0, \overline{S}_i^1, \dots$ . We denote by  $\Pi_{i \rightarrow j}$  the concatenation of the messages sent by player  $i$  to player  $j$  during the course of the protocol. The transcript of the (whole) protocol, denoted  $\Pi$ , is obtained by concatenating all the  $\Pi_i$  ordered by, say, player index.

We will give most of the definitions for the case where all functions  $f_i$  are the same function, that we denote by  $f$ . The definitions in the case of family of functions are similar.

**Definition 3.1.** For  $\epsilon \geq 0$ , a protocol  $\pi$   $\epsilon$ -computes a function  $f$  if for all  $(x_1, \dots, x_k) \in \mathcal{X}_1 \times \dots \times \mathcal{X}_k$ :

1. For all possible assignments for the random sources  $R_i$ ,  $1 \leq i \leq k$ , and  $R^p$ , every player eventually (i.e., in finite time) writes on its output tape (a non-empty string).
2. With probability at least  $1 - \epsilon$  (over all random sources) the following event occurs: each player  $i$  writes on its output tape the value  $f(x)$ , i.e., the correct value of the function.

For simplicity we also assume that a protocol must eventually stop. That is, for all possible inputs and all possible assignments for the random sources, eventually (i.e., in finite time) there is no message in transit.

---

<sup>2</sup>We require that each player writes only once on its output tape so that the local view of the player determines the local output of the protocol (i.e., so that players itself “knows” the output). This requirement is needed since a player may not know locally that the protocol ended.



## 3.2 Comparison to other models

The somewhat restricted model (compared to the general asynchronous model) that we work with allows us to define a measure similar to information cost that we will later show to have desirable properties and to be of use. Notice that the general asynchronous model is problematic in this respect since one bit of communication can bring  $\log(k)$  bits of information, as not only the content of the message but also the identity of the sender may reveal information. Thus, information cannot be used as a lower bound on communication. In our case, the sets  $S_i^l$  and  $\overline{S}_i^l$  are determined by the current view of the player,  $(\Pi_i)$  contains only the content of the messages, and thus the desirable relation between the communication and the information is maintained. On the other hand, our restriction is natural, does not seem to be very restrictive (practically all protocols in the literature adhere to our model), and does not exclude the existence of private protocols.

To exemplify the above mentioned issue in the general asynchronous model consider the following simple example of a deterministic protocol, for 4 players  $A, B$  and  $C, D$ , which allows  $A$  to transmit to  $B$  its input bit  $x$ , but where all messages sent in the protocol are the bit 0, and the protocol generates only a single transcript over all possible inputs.

**A:** If  $x = 0$  send 0 to  $C$ ; after receiving 0 from  $C$ , send 0 to  $D$ .

    If  $x = 1$  send 0 to  $D$ ; after receiving 0 from  $D$ , send 0 to  $C$

**B:** After receiving 0 from a player, send 0 back to that player.

**C,D:** After receiving 0 from  $A$  send 0 to  $B$ . After receiving 0 from  $B$  send 0 to  $A$ .

It is easy to see that  $B$  learns the value of  $x$  from the order of the messages it gets.

There has been a long series of works about multi-party communication protocols in different variants of models, for example [CKS03, Gro09, Jay09, PVZ12, CRR14, CR15]. In [BEO<sup>+</sup>13], Braverman et al. consider a restricted class of protocols working in the *coordinator model*: an additional player with no input can communicate privately with each player, and the players can only communicate with the coordinator.

We first note that the coordinator model does not yield exact bounds for the multi-party communication complexity in the peer-to-peer model (neither in our model nor in the most general one). Namely, a protocol in the peer-to-peer model can be transformed into a protocol in the coordinator model with an  $O(\log k)$  multiplicative factor in the communication complexity, by sending any message to the coordinator with a  $O(\log k)$ -bit label indicating its destination. This factor is sometimes necessary, e.g., for the  $q$ -**index** function, where players  $P_i$ ,  $0 \leq i \leq k-1$ , each holds an input bit  $x_i$ , player  $P_k$  holds  $q$  indices  $0 \leq j_\ell \leq k-1$ ,  $1 \leq \ell \leq q$ , and  $P_k$  should learn the vector  $(x_{j_1}, x_{j_2}, \dots, x_{j_q})$ : in the coordinator model the communication complexity of this function is  $\Theta(\min\{k, q \log k\})$ , while in both peer-to-peer models there is a protocol for this function that sends only (at most)  $\min\{k, 2q\}$  bits, where  $P_k$  just queries the appropriate other players. But this multiplicative factor between the complexities in the two models is not always necessary: the communication complexity of the parity function **Par** is  $\Theta(k)$  both in the peer-to-peer models and in the coordinator model.

Moreover, when studying private protocols in the peer-to-peer model, the



coordinator model does not offer any insight. In the (asynchronous) coordinator model, described in [DF89] and used for instance in [BEO<sup>+</sup>13], if there is no privacy requirement with respect to the coordinator, it is trivial to have a private protocol by all players sending their input to the coordinator, and the coordinator returning the results to the players. If there is a privacy requirement with respect to the coordinator, then if there is a random source shared by all the players (but not the coordinator), privacy is always possible using the protocol of [FKN94]. If no such source exists, privacy is impossible in general. This follows from the results of Braverman et al. [BEO<sup>+</sup>13] who show a non-zero lower bound on the total internal information complexity of all parties (including the coordinator) for the function *Disjointness* in that model.

Note also that the private protocols described in [BOGW88, CCD88] (and further work) are defined in the synchronous setting, and thus can be adapted to our communication model (the sets  $\overline{S}_i^j$  and  $S_i^j$  are always all the players and hence even independent of the current views).

In the sequel we also use a special case of our model, where the sets  $\overline{S}_i^j$  and  $S_i^j$  are a function only of  $i$  and  $j$ , and not of the entire current view of the player. This is a natural special case for protocols which we call *oblivious protocols*, where the communication pattern is fixed and is not a function of the input or randomness. Clearly, the messages themselves remain a function of the view of the players. This model also allows for private protocols.

### 3.3 Communication complexity and information complexity

Communication complexity, introduced in [Yao79], measures how many bits of communication are needed in order for a set of players to compute with error  $\epsilon$  a given function of their inputs. The allowed error  $\epsilon$ , implicit in many of the contexts, will be written explicitly as a superscript when necessary.

**Definition 3.2.** *The communication cost of a protocol  $\pi$ ,  $CC(\pi)$ , is the maximal length of the transcript of  $\pi$  over all possible inputs, private randomness and public randomness.*

**Definition 3.3.**  *$CC(f)$  denotes the communication cost of the best protocol computing  $f$ .*

Information complexity measures the amount of information that must be transmitted so that the players can compute a given function of their joint inputs. One of its main uses is to provide a lower bound on the communication complexity of the function. In the two-party setting, this measure led to interesting results on the communication complexity of various functions, such as *AND* and *Disjointness*. We now focus on designing an analogue to the information cost, for the multi-party setting. The notion of internal information cost for two-party protocols (c.f. [CSWY01, BYJKS02, Bra12]) can be easily generalized to any number of players:

**Definition 3.4.** *The internal information cost of a protocol  $\pi$  for  $k$  players, with respect to input distribution  $\mu$ , is the sum of the information revealed to each player about the inputs of the other players:  $IC_\mu(\pi) = \sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i R_i R^p)$ .*

Intuitively, the information cost of a protocol is the amount of information each player learns about the inputs of the other players during the protocol. The definition we give above, when restricted to two players is the same as in [Bra12], even though they look slightly different. This is because we explicit the role of the randomness, which will allow us to later bound the amount of randomness needed for private protocols in the multi-party setting.

The *internal information complexity* of a function  $f$  with respect to input distribution  $\mu$ , as well as the *internal information complexity* of a function  $f$ , can be defined for the multi-party case based on the information cost of a protocol, just as in the 2-party case.

**Definition 3.5.** *The internal information complexity of a function  $f$ , with respect to input distribution  $\mu$  is the infimum of the internal information cost over all protocols computing  $f$  on input distribution  $\mu$ :*

$$\text{IC}_\mu(f) = \inf_{\pi \text{ computing } f} \text{IC}_\mu(\pi).$$

**Definition 3.6.** *The internal information complexity of a function  $f$  is the infimum, over all protocols  $\pi$  computing  $f$ , of the information cost of  $\pi$  when run on the worst input distribution for that protocol:*

$$\text{IC}(f) = \inf_{\pi \text{ computing } f} \sup_{\mu} \text{IC}_\mu(\pi).$$

**Proposition 3.7** ([BR11]). *For any protocol  $\pi$  and input distribution  $\mu$ ,  $\text{CC}(\pi) \geq \text{IC}_\mu(\pi)$ . Thus, for any function  $f$ ,  $\text{CC}(f) \geq \text{IC}(f)$ .*

The information revealed to a given player by a protocol can be written in several ways:

**Proposition 3.8.** *For any protocol  $\pi$ , for any player  $i$ :*

$$\begin{aligned} I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R^p) &= I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p) = I(X_{-i}; \Pi_i \mid X_i R_i R^p), \text{ and} \\ I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R^p f(X)) &= I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p f(X)) = I(X_{-i}; \Pi_i \mid X_i R_i R^p f(X)). \end{aligned}$$

*Proof.* We only prove the first statement, since the two statements have similar proofs.

We first prove the equality  $I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p) = I(X_{-i}; \Pi_i \mid X_i R_i R^p)$ . We write  $\overleftarrow{\Pi_i}$  as a sequence  $b_0 \dots b_q$  of bits, and define  $\overleftarrow{\Pi_i}^{<j}$  to be the set of bits with index less than  $j$  and *received* by player  $i$ . Similarly,  $\overrightarrow{\Pi_i}^{<j}$  is defined to be the set of bits with index less than  $j$  and *sent* by player  $i$ .

$$\begin{aligned} I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p) &= I(X_{-i}; b_0 \dots b_q \mid X_i R_i R^p) \\ &= \sum_{j \geq 0} I(X_{-i}; b_j \mid X_i R_i R^p b_0 \dots b_{j-1}) \text{ (using chain rule (proposition 2.6).)} \end{aligned}$$

For any  $j$  and any bit  $b_j \in \overleftarrow{\Pi_i}$  that is sent in local round  $\ell$ , bit  $b_j$  is determined by  $D_i^{\ell-1}$ . Therefore, for such  $b_j$ ,  $H(b_j \mid X_i R_i R^p b_0 \dots b_{j-1}) = 0$  and thus  $I(X_{-i}; b_j \mid X_i R_i R^p b_0 \dots b_{j-1}) = 0$ . Hence

$$I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p) = \sum_{j \mid b_j \in \Pi_i} I(X_{-i}; b_j \mid X_i R_i R^p b_0 \dots b_{j-1}).$$

Fix  $j$  such that  $b_j$  is received by player  $i$ . We now show that

$$I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j} \overrightarrow{\Pi_i^{<j}}) = I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j}).$$

Since  $H(\overrightarrow{\Pi_i^{<j}} \mid X_i R_i R^p \Pi_i^{<j}) = 0$ , we have  $I(X_{-i}; \overrightarrow{\Pi_i^{<j}} \mid X_i R_i R^p \Pi_i^{<j}) = 0$  and applying lemma 2.9,

$$I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j} \overrightarrow{\Pi_i^{<j}}) \geq I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j}).$$

Since  $H(\overrightarrow{\Pi_i^{<j}} \mid X_i R_i R^p b_j \Pi_i^{<j}) = 0$ , we have  $I(X_{-i}; \overrightarrow{\Pi_i^{<j}} \mid X_i R_i R^p b_j \Pi_i^{<j}) = 0$  and applying lemma 2.8,

$$I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j} \overrightarrow{\Pi_i^{<j}}) \leq I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j}).$$

We get:

$$\begin{aligned} I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i) &= \sum_{j \mid b_j \in \Pi_i} I(X_{-i}; b_j \mid X_i R_i R^p b_0 \dots b_{j-1}) \\ &= \sum_{j \mid b_j \in \Pi_i} I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j} \overrightarrow{\Pi_i^{<j}}) \\ &= \sum_{j \mid b_j \in \Pi_i} I(X_{-i}; b_j \mid X_i R_i R^p \Pi_i^{<j}) \\ &= I(X_{-i}; \Pi_i \mid X_i R_i R^p) \text{ (using chain rule (proposition 2.6).)} \end{aligned}$$

We now prove the equality  $I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R^p) = I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p)$ .

We denote by  $\overleftarrow{\Pi_i^{<j}} = b_0 \dots b_{j-1}$ . We have:

$$\begin{aligned} I(X_{-i}; \overleftarrow{\Pi_i} \mid X_i R_i R^p) &= I(X_{-i}; b_0 \dots b_q \mid X_i R_i R^p) \\ &= \sum_j I(X_{-i}; b_j \mid X_i R_i R^p \overleftarrow{\Pi_i^{<j}}) \text{ (using chain rule (proposition 2.6).)} \end{aligned}$$

We show that  $I(X_{-i}; R_i \mid X_i R^p \overleftarrow{\Pi_i^{<j}}) = 0$ . By proposition 2.4, it is equivalent to show that for each possible value of  $(X_i, R^p, \overleftarrow{\Pi_i^{<j}})$ , the random variables  $X_{-i}$  and  $R_i$  are independent. To do this, we fix a value  $(x_i, r, \overleftarrow{\pi_i^{<j}})$  of  $(X_i, R^p, \overleftarrow{\Pi_i^{<j}})$ , take a value  $x_{-i}$  of  $X_{-i}$  such that the event  $X_{-i} = x_{-i}$  is compatible with the event  $(X_i, R^p, \overleftarrow{\Pi_i^{<j}}) = (x_i, r, \overleftarrow{\pi_i^{<j}})$ , and prove that for any possible value  $r_i$  of  $R_i$ , the quantity  $\Pr[r_i \mid x_i \text{ } r \text{ } \overleftarrow{\pi_i^{<j}} \text{ } x_{-i}]$  does not depend on  $x_{-i}$ .

We have

$$\Pr[r_i \mid x_i \text{ } r \text{ } \overleftarrow{\pi_i^{<j}} \text{ } x_{-i}] = \sum_{r_{-i}} \Pr[r_i \mid x_i \text{ } r \text{ } \overleftarrow{\pi_i^{<j}} \text{ } x_{-i} r_{-i}] \Pr[r_{-i} \mid x_i \text{ } r \text{ } \overleftarrow{\pi_i^{<j}} \text{ } x_{-i}].$$

We define the set function

$\rho : (x_i, r \text{ } \overleftarrow{\pi_i^{<j}}, x_{-i}, r_{-i}) \mapsto \{r_i \mid \overleftarrow{\Pi_i^{<j}}(x_i, x_{-i}, r, r_i, r_{-i}) = \overleftarrow{\pi_i^{<j}}\}$ , i.e., the set of possible values for  $r_i$ , such that inputs  $x_i, x_{-i}$ , public randomness  $r$  and private randomness  $r_i, r_{-i}$  will lead to a transcript of player  $i$  beginning by  $\overleftarrow{\pi_i^{<j}}$ .

Note that  $\Pr[r_{-i} \mid x_i \text{ } r \text{ } \overleftarrow{\pi_i^{<j}} \text{ } x_{-i}] \neq 0 \iff \rho(x_i, r \text{ } \overleftarrow{\pi_i^{<j}}, x_{-i}, r_{-i}) \neq \emptyset$ .

Also note that  $r_i \notin \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i}) \iff \Pr[r_i \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i} r_{-i}] = 0$  and that

$$\forall r_i \in \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i}), \Pr[r_i \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i} r_{-i}] = \frac{1}{|\rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i})|}.$$

We prove that the function  $\rho(x_i, r, \overleftarrow{\pi_i}^{<j}, \cdot, \cdot)$  is constant on the set  $\{(x'_{-i}, r'_{-i}) \mid \Pr[r'_{-i} \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x'_{-i}] \neq 0\}$ . Let  $r_{-i}$  such that  $\Pr[r_{-i} \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i}] \neq 0$  and  $(x'_{-i}, r'_{-i})$  such that  $\Pr[r'_{-i} \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x'_{-i}] \neq 0$ . Let  $r_i \in \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i})$  and  $r'_i \in \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x'_{-i}, r'_{-i})$ . We get that  $\overleftarrow{\Pi_i}^{<j}(x_i, r, x'_{-i}, r_i, r'_{-i}) = \overleftarrow{\pi_i}^{<j}$ : this comes from the facts that  $\overleftarrow{\Pi_i}^{<j}(x_i, r, x_{-i}, r_i, r_{-i}) = \overleftarrow{\pi_i}^{<j} = \overleftarrow{\Pi_i}^{<j}(x_i, r, x'_{-i}, r'_i, r'_{-i})$  and from the fact that the action of the players are based their current view. Thus  $r_i \in \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x'_{-i}, r'_{-i})$ . For the same reason,  $r'_i \in \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i})$ .

We have shown that  $\rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i}) = \rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x'_{-i}, r'_{-i})$ .

Let  $\alpha(x_i, r, \overleftarrow{\pi_i}^{<j})$  be the constant value of the function  $|\rho(x_i, r, \overleftarrow{\pi_i}^{<j}, \cdot, \cdot)|$ .

$$\begin{aligned} \Pr[r_i \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i}] &= \sum_{r_{-i}} \Pr[r_i \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i} \ r_{-i}] \Pr[r_{-i} \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i}] \\ &= \sum_{r_{-i}} \frac{1}{|\rho(x_i, r, \overleftarrow{\pi_i}^{<j}, x_{-i}, r_{-i})|} \Pr[r_{-i} \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i}] \\ &= \sum_{r_{-i}} \frac{1}{\alpha(x_i, r, \overleftarrow{\pi_i}^{<j})} \Pr[r_{-i} \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i}] \\ &= \frac{1}{\alpha(x_i, r, \overleftarrow{\pi_i}^{<j})} \end{aligned}$$

which is independent of  $x_{-i}$ . Thus  $\Pr[r_i \mid x_i \ r \ \overleftarrow{\pi_i}^{<j} \ x_{-i}] = \Pr[r_i \mid x_i \ r \ \overleftarrow{\pi_i}^{<j}]$ . We have shown that the random variables  $X_{-i}$  and  $R_i$  are independent conditioned on  $(X_i, R^p, \overleftarrow{\Pi_i}^{<j})$ , and thus  $I(X_{-i}; R_i \mid X_i R^p \overleftarrow{\Pi_i}^{<j}) = 0$ . By lemma 2.9, this implies

$$I(X_{-i}; b_j \mid X_i R^p R_i \overleftarrow{\Pi_i}^{<j}) \geq I(X_{-i}; b_j \mid X_i R^p \overleftarrow{\Pi_i}^{<j}).$$

A similar reasoning, along with lemma 2.8, leads to

$$I(X_{-i}; b_j \mid X_i R^p R_i \overleftarrow{\Pi_i}^{<j}) \leq I(X_{-i}; b_j \mid X_i R^p \overleftarrow{\Pi_i}^{<j}).$$

We have:

$$\begin{aligned} I(X_{-i}; \overleftarrow{\pi_i} \mid X_i R^p R_i) &= \sum_j I(X_{-i}; b_j \mid X_i R_i R^p \overleftarrow{\Pi_i}^{<j}) \\ &= \sum_j I(X_{-i}; b_j \mid X_i R^p \overleftarrow{\Pi_i}^{<j}) \\ &= I(X_{-i}; \overleftarrow{\pi_i} \mid X_i R^p) \text{ (using chain rule (proposition 2.6).)} \end{aligned}$$

⌋

### 3.4 Information complexity and privacy

The definition of a *private protocol* as defined in [BOGW88, CCD88] is the following.

**Definition 3.9.** A  $k$ -player protocol  $\pi$  for computing a family of functions  $(f_i)$  is *private*<sup>3</sup> if for every player  $i \in \llbracket 1, k \rrbracket$ , for all pair of inputs  $x = (x_1, \dots, x_k)$  and  $x' = (x'_1, \dots, x'_k)$ , such that  $f_i(x) = f_i(x')$  and  $x_i = x'_i$ , for all possible private random tapes  $r_i$  of player  $i$ , and all possible public random tapes  $r^p$ , it holds that for any transcript  $T$

$$\Pr[\Pi_i = T \mid R_i = r_i ; X = x ; R^p = r^p] = \Pr[\Pi_i = T \mid R_i = r_i ; X = x' ; R^p = r^p]$$

where the probability is over the randomness  $R_{-i}$ .

The notion of privacy has an equivalent formulation in terms of information.

**Proposition 3.10.** A protocol  $\pi$  is private if and only if for all input distributions  $\mu$ ,

$$\sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X)) = 0.$$

*Proof.* By proposition 2.4, definition 3.9 is equivalent to the following:

$$\forall i, I(X_{-i}; \Pi_i \mid X_i R_i R^p f(X)) = 0.$$

Since  $I$  is non-negative, this is equivalent to

$$\sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X)) = 0.$$

□

It is well known that in the multi-party number-in-the-hand peer-to-peer setting (for  $k \geq 3$ ), unlike in the two-party case, any function can be privately computed.

**Theorem 3.11** ([BOGW88],[CCD88]). *Any family of functions of more than two variables can be computed by a private protocol.*

Using the above theorem, we can give the following lemma.

**Lemma 3.12.** *For any family of functions  $(f_i)$  of more than two variables and any  $\mu$ ,*

$$\text{IC}_\mu(f) \leq \sum_{i=1}^k H(f_i(X)), \text{ where } X \text{ is distributed according to } \mu.$$

---

<sup>3</sup>In this paper we consider only the setting of 1-privacy, which we call here for simplicity, privacy.

*Proof.* Let  $\pi$  be a  $k$ -player private protocol computing  $(f_i)$ . Fix a distribution  $\mu$  on the inputs.

$$\begin{aligned}
\text{IC}_\mu(\pi) &= \sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^k I(X_{-i}; \Pi_i f_i(X) \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^k [I(X_{-i}; f_i(X) \mid X_i R_i R^p) + I(X_{-i}; \Pi_i \mid X_i R_i R^p f_i(X))] \\
&\leq \sum_{i=1}^k I(X_{-i}; f_i(X) \mid X_i R_i R^p) \\
&\leq \sum_{i=1}^k H(f_i(X)).
\end{aligned}$$

$$\text{Now, } \text{IC}_\mu(f) \leq \text{IC}_\mu(\pi) \leq \sum_{i=1}^k H(f_i(X)).$$

┘

This lemma shows that IC cannot be used in the multi-party setting for any meaningful lower bounds on the communication complexity, since its value is always upper bounded by the entropies of the functions. Our goal is to get lower bounds tight in both  $k$  and  $n$ . For this reason, we introduce a new information theoretic quantity for the multi-party setting.

## 4 The new measure: Public Information Cost

We now introduce a new information theoretic quantity which can be used instead of IC in the multi-party setting. The notion we define will be suitable for studying multi-party communication in a model which is only a slight restriction on the general asynchronous model, and which allows for private protocols. This means that while IC will be at most the entropies of the functions, our new notion remains a strong lower bound for communication.

**Definition 4.1.** For any  $k$ -player protocol  $\pi$  and any input distribution  $\mu$ , we define the public information cost of  $\pi$ :

$$\text{PIC}_\mu(\pi) = \sum_{i=1}^k I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p).$$

The difference between PIC and IC is the presence of the other parties private coins,  $R_{-i}$ , in the formula. If  $\pi$  is a protocol using only public randomness, then for any input distribution  $\mu$ ,  $\text{PIC}_\mu(\pi) = \text{IC}_\mu(\pi)$ , and hence the name public information cost.

The public information cost measures both the information about the inputs learned by the players and the information that is hidden by the use of private coins. It can be decomposed, using the chain rule, into two terms, making

explicit the contribution of the internal information cost and of the private randomness of the players.

**Proposition 4.2.** *For any  $k$ -player protocol  $\pi$  and any input distribution  $\mu$ ,*

$$\text{PIC}_\mu(\pi) = \text{IC}_\mu(\pi) + \sum_{i=1}^k I(R_{-i}; X_{-i} | X_i \Pi_i R_i R^p).$$

The meaning of the second term is the following. At the end of the protocol, player  $i$  knows its input  $X_i$ , its private coins  $R_i$ , the public coins  $R^p$  and its transcript  $\Pi_i$ . Suppose that the private randomness  $R_{-i}$  of the other players is now revealed to player  $i$ . This brings to it some new information  $I(R_{-i}; X_{-i} | X_i \Pi_i R_i R^p)$  about the inputs  $X_{-i}$  of the other players.

We also define the public information complexity of a function.

**Definition 4.3.** *For any function  $f$  and any input distribution  $\mu$ , we define the quantity*

$$\text{PIC}_\mu(f) = \inf_{\pi \text{ computing } f} \text{PIC}_\mu(\pi).$$

**Definition 4.4.** *For any  $f$ , we define the quantity  $\text{PIC}(f) = \inf_{\pi \text{ computing } f} \sup_{\mu} \text{PIC}_\mu(\pi)$ .*

The public information cost is a lower bound for the communication complexity.

**Proposition 4.5.** *For any protocol  $\pi$  and input distribution  $\mu$ ,  $\text{CC}(\pi) \geq \text{PIC}_\mu(\pi)$ . Thus, for any function  $f$ ,  $\text{CC}(f) \geq \text{PIC}(f)$ .*

*Proof.*

$$\begin{aligned} \text{PIC}_\mu(\pi) &= \sum_{i=1}^k I(X_{-i}; R_{-i} | X_i R_i R^p) + I(X_{-i}; \Pi_i | X_i R R^p) \text{ (by the chain rule)} \\ &= \sum_{i=1}^k I(X_{-i}; \Pi_i | X_i R R^p) \text{ (since the first term was 0 by proposition 2.4)} \\ &= \sum_{i=1}^k H(\Pi_i | X_i R R^p) \text{ (since } H(\Pi_i | X R R^p) = 0 \text{)} \\ &\leq \sum_{i=1}^k H(\Pi_i) \text{ (by proposition 2.5.)} \end{aligned}$$

Using proposition 2.2, for each  $i$ ,  $H(\Pi_i)$  is upper bounded by the expected size of  $\Pi_i$ . As the expected size of  $\Pi$  is upper bounded by the sum over  $i$  of the expected size of  $\Pi_i$ , we get  $\text{CC}(\pi) \geq \text{PIC}_\mu(\pi)$ . ┘

In fact, as we show below, the public information cost of a function is equal to its internal information cost in a setting where only public randomness is allowed. The role of private coins in communication protocol has been studied for example in [BG14, BBK<sup>+</sup>13, Koz15]. In the next section we will see that the difference between the public information cost and the information cost is related to the private coins used during the protocol.



**Theorem 4.6.** For any function  $f$  and input distribution  $\mu$ ,

$$\text{PIC}_\mu(f) = \inf_{\pi \text{ computing } f, \text{ using only public coins}} \text{IC}_\mu(\pi)$$

and

$$\text{PIC}(f) = \inf_{\pi \text{ computing } f, \text{ using only public coins}} \sup_{\mu} \text{IC}_\mu(\pi).$$

*Proof.* It suffices to show than one can turn any protocol  $\pi$  into a public coin protocol  $\pi'$  such that for all input distributions  $\mu$ ,  $\text{PIC}_\mu(\pi') = \text{PIC}_\mu(\pi)$ .

Fix a protocol  $\pi$  and an input distribution  $\mu$ . Let  $R_i$  denote the private randomness of player  $i$  in  $\pi$ , and  $R = (R_i)$ . Define  $\pi'$ , where the players act as they do in  $\pi$ , but use public randomness instead of their private randomness whenever they need to use it. For this, split the random tape into two sub-tapes  $R = (R_i)$ , to be used instead of the private randomness of each player, and  $R^p$ , to be used as public randomness. We have

$$\begin{aligned} \text{PIC}_\mu(\pi') &= \sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i R R^p) \\ &= \sum_{i=1}^k [I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p) - I(X_{-i}; R_{-i} \mid X_i R_i R^p)] \quad (\text{chain rule}) \\ &= \sum_{i=1}^k I(X_{-i}; \Pi_i R_{-i} \mid X_i R_i R^p) \quad (\text{since the second term was 0}) \\ &= \text{PIC}_\mu(\pi). \end{aligned}$$

□

The following property of the public information cost will be useful for zero-error protocols.

**Proposition 4.7.** For any function  $f$ , for any input distribution  $\mu$ ,  $\text{PIC}_\mu^0(f) = \text{IC}_\mu^{\text{det}}(f)$  where

$$\text{IC}_\mu^{\text{det}}(f) = \inf_{\pi \text{ deterministic protocol computing } f} \text{IC}_\mu(\pi).$$

*Proof.* Let  $\pi$  be a zero-error protocol for  $f$ . By theorem 4.6, one can assume that  $\pi$  has no private randomness.

$$\begin{aligned} \text{IC}_\mu(\pi) &= \sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i R^p) \\ &= \sum_{i=1}^k \mathbb{E}_r [I(X_{-i}; \Pi_i \mid X_i, R^p = r)] \\ &= \mathbb{E}_r \left[ \sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i, R^p = r) \right]. \end{aligned}$$

Letting  $t(r) = \sum_{i=1}^k I(X_{-i}; \Pi_i \mid X_i, R^p = r)$ , it holds  $\text{IC}_\mu(\pi) = \mathbb{E}_r [t(r)]$ . Let  $r_0$  be a value of the public random tape minimizing the function  $t$ , and define  $\pi^0$  as the

protocol behaving like  $\pi$  on the random tape  $r_0$ . Note that  $\pi^0$  is a deterministic zero-error protocol computing  $f$ . We now prove that  $\text{IC}_\mu(\pi^0) \leq \text{IC}_\mu(\pi)$ , which concludes the proof, by using the definition of  $\pi^0$ .

$$\text{IC}_\mu(\pi^0) = \sum_{i=1}^k I(X_{-i}; \Pi_i^0 | X_i) = \sum_{i=1}^k I(X_{-i}; \Pi_i | X_i R = r_0) \leq \text{IC}_\mu(\pi).$$

┘

PIC and IC are strictly different even in the two party case. We prove below that for the AND function, the public information cost is  $\log 3 \simeq 1.58$ , while, as shown in [BGPW13],  $\text{IC}^0(\text{AND}) \simeq 1.49$ . This implies that the protocol that achieves the optimal information cost for AND must use private coins. We remark also that in [BGPW13] it is shown that the external information cost of AND, that we do not consider here, is  $\log(3)$ .

**Proposition 4.8.** *For two players,  $\text{PIC}^0(\text{AND}) = \log(3) \simeq 1.58$ .*

*Proof.* Let  $\pi$  be a zero-error protocol for AND. By proposition 4.7, we can assume that  $\pi$  is deterministic. Suppose player 1 is speaking first.  $\pi$  being deterministic, player 1 is either sending his bit  $x$  or sending  $1 - x$ . Player 2 is then able to compute the value of AND, and as player 1 must be able to compute AND at the end of the protocol, the optimal protocol consists in player 2 sending back the value of AND to player 1. We compute the value of  $\text{PIC}_\mu(\pi)$ , for  $\mu$  defined as follows:  $X \sim \text{Ber}(\alpha, 1 - \alpha)$  and  $Y \sim \text{Ber}(\beta, 1 - \beta)$ .

$$\begin{aligned} \text{PIC}_\mu(\pi) &= I(X; \Pi | Y) + I(Y; \Pi | X) \\ &= H(X) + [\alpha I(Y; \Pi | X = 0) + (1 - \alpha) I(Y; \Pi | X = 1)]. \end{aligned}$$

When  $X = 1$ , player 1 will learn the value of  $Y$ , as  $\text{AND}(X, Y) = Y$ . Thus  $\text{PIC}_\mu(\pi) = H(X) + (1 - \alpha)H(Y)$ . For any  $\alpha$ , this quantity is maximal for  $\beta = \frac{1}{2}$ .

Thus we study the function

$f(\alpha) = \log(\alpha) + (\alpha - 1) \log(1 - \alpha) + 1 - \alpha$  for  $\alpha \in [0, 1]$ .  $f$  is continuous on  $[0, 1]$  and differentiable on  $]0, 1[$ . We have:

$$f'(\alpha) = -\log(\alpha) - 1 + \log(1 - \alpha) + 1 - 1 = \log\left(\frac{1}{\alpha} - 1\right) - 1.$$

$f'$  is continuous and decreasing on  $]0, 1[$ , and admits the unique root  $\frac{1}{3}$ .  $\text{PIC}_\mu(\pi)$

is thus maximized for  $\alpha = \frac{1}{3}$ , its value being  $f(\alpha) = \log(3)$ .

┘

## 5 Private computation, randomness, and PIC

We have seen that the public information cost of a function is equal to the information cost of the function when we only consider public coin protocols, and that in order to decrease the information cost even further, the players must use private randomness. We will see now that the difference between the public information cost of a protocol and its information cost can provide a lower bound on the amount of private randomness the players use during the protocol. The entropy of the transcript of the protocol, conditioned on the

inputs and the public coins, is defined as  $H(\Pi \mid XR^p)$ . Once the input and the public coins are fixed, the entropy of the transcript of the protocol comes solely from the private randomness. Thus it provides a lower bound on the entropy of the private randomness used by the players.

**Theorem 5.1.** *Let  $f = (f_i)$  be a family of functions of  $k$  variables. Let  $\pi$  be a protocol for  $f$ . For any input distribution  $\mu$ , it holds:*

$$H_\mu(\Pi \mid XR^p) \geq \frac{\text{PIC}_\mu(\pi) - \text{IC}_\mu(\pi)}{k-1}.$$

Thus running a protocol for  $f$  with information cost  $I_\mu$  requires entropy

$$H_\mu(\Pi \mid XR^p) \geq \frac{\text{PIC}_\mu(f) - I_\mu}{k-1}.$$

*Proof.* Fix  $\mu$  a distribution on inputs.

$$\begin{aligned} \text{PIC}_\mu(\pi) &= \text{IC}_\mu(\pi) + \sum_{i=1}^k I(R_{-i}; X_{-i} \mid X_i R_i R^p \Pi_i) \quad (\text{proposition 4.2}) \\ &\leq \text{IC}_\mu(\pi) + \sum_{i=1}^k I(R_{-i}; X R_i \Pi_i \mid R^p). \end{aligned}$$

For any  $i$ ,

$$\begin{aligned} I(R_{-i}; X R_i \Pi_i \mid R^p) &\leq I(R_{-i}; X R_i \Pi \mid R^p) \\ &\leq I(R_{-i}; \Pi \mid R^p) + I(R_{-i}; X R_i \mid \Pi R^p) \\ &\leq I(R_{-i}; \Pi \mid R^p) + I(R_{-i}; X \mid \Pi R^p) \quad (\text{as } I(R_{-i}; R_i \mid X \Pi R^p) = 0, \text{ cf. lemma A.1}) \\ &\leq I(R_{-i}; \Pi X \mid R^p) \\ &= \sum_{j \neq i} I(R_j; \Pi X \mid R_{<j, \neq i} R^p) \\ &\leq \sum_{j \neq i} I(R_j; \Pi X \mid R^p) \quad (\text{by lemma 2.8 with } I(R_j; R_{<j, \neq i} \mid X \Pi R^p) = 0.) \end{aligned}$$

Thus

$$\begin{aligned} \text{PIC}_\mu(\pi) &\leq \text{IC}_\mu(\pi) + (k-1) \sum_{i=1}^k I(R_i; \Pi X \mid R^p) \\ &\leq \text{IC}_\mu(\pi) + (k-1) I(R; \Pi X \mid R^p) \quad (\text{by lemma 2.9}) \\ &\leq \text{IC}_\mu(\pi) + (k-1) I(R; \Pi \mid X R^p) \quad (\text{as } I(R; X \mid R^p) = 0) \\ &\leq \text{IC}_\mu(\pi) + (k-1) H_\mu(\Pi \mid X R^p). \end{aligned}$$

┘

Using lemma 3.12, we can bound the randomness required to run a private protocol.

**Corollary 5.2.** *Let  $f = (f_i)$  be a family of functions of  $k$  variables. Let  $\pi$  be a  $k$ -party private protocol for  $f$ . For any distribution  $\mu$  on inputs,*

$$H_\mu(\Pi \mid X R^p) \geq \frac{\text{PIC}_\mu(f) - \sum_{i=1}^k H_\mu(f_i)}{k-1}.$$

## 6 A direct sum for PIC ?

The *direct sum* property is a fundamental question in complexity theory, and has been studied for many computation models. A direct sum theorem affirms that the amount of resources needed to perform  $t$  independent tasks is at least the sum of the resources needed to perform each of the  $t$  tasks. In this section we show that a direct sum property for PIC implies a direct sum property for CC. For this, we prove a compression result by extending [BBK<sup>+</sup>13, Pan15] to the multi-party case. Note that information complexity has a direct sum property in the multi-party case. For PIC, it is easy to prove the following inequality.

**Theorem 6.1.** *For any  $k$ -variable functions  $f$  and  $g$ , for any distribution  $\mu$  on inputs of  $f$ , for any distribution  $\eta$  on inputs of  $g$ , we have*

$$\text{PIC}_{\mu \times \eta}(f \times g) \leq \text{PIC}_{\mu}(f) + \text{PIC}_{\eta}(g).$$

In order to understand whether the opposite inequality holds, i.e., whether a direct sum property holds for PIC, we first need to study the problem of compressing communication.

### 6.1 Relation between PIC and CC: A compression result

An important open question is how well we can compress the communication cost of an interactive protocol. Compression results have appeared in [BBCR10, BR11, BBK<sup>+</sup>13, Pan15, BMY15], while, on the other hand, [GKR14, GKR15b, RS15, FJK<sup>+</sup>15, GKR15a] focus on the hardness of compressing communication protocols. Here, we present a compression result with regards to the average-case communication complexity and the public information cost.

**Definition 6.2.** *The average-case communication complexity of a protocol  $\pi$  with respect to the input distribution  $\mu$ , denoted  $\text{ACC}_{\mu}(\pi)$ , is the expected number of bits that are transmitted in an execution of  $\pi$  for inputs distributed according to  $\mu$  and uniform randomness.*

**Theorem 6.3.** *Suppose there exists a protocol  $\pi$  to compute a  $k$ -variable function  $f$  over the distribution  $\mu$  with error probability  $\epsilon$ . Then there exists a public-coin protocol  $\rho$  that computes  $f$  over  $\mu$  with error  $\epsilon + \delta$ , and with average communication complexity*

$$\text{ACC}_{\mu}(\rho) = \mathcal{O} \left( k^2 \text{PIC}_{\mu}(\pi) \log(\text{CC}(\pi)) \left( \log(k \text{CC}(\pi)) + \log \frac{k^2 \text{PIC}_{\mu}(\pi) \log(\text{CC}(\pi))}{\delta} \right) \right).$$

The proof of the above theorem will follow from extending the compression result presented in [BBK<sup>+</sup>13, Pan15] to the case of  $k > 2$  players.

**Theorem 6.4.** *Suppose there exists a public coin protocol  $\pi$  to compute a  $k$ -variable function  $f$  over the distribution  $\mu$  with error probability  $\epsilon$ . Then there exists a public-coin protocol  $\rho$  that computes  $f$  over  $\mu$  with error  $\epsilon + \delta$ , and with average communication complexity*

$$\text{ACC}_{\mu}(\rho) = \mathcal{O} \left( k^2 \text{IC}_{\mu}(\pi) \log(\text{CC}(\pi)) \left( \log(k \text{CC}(\pi)) + \log \frac{k^2 \text{IC}_{\mu}(\pi) \log(\text{CC}(\pi))}{\delta} \right) \right).$$

Theorems 6.4 and 4.6, which make the link between the public information cost of general protocols and the information cost of public coins protocol, imply theorem 6.3.

In the two-party compression scheme of [BBK<sup>+</sup>13, Pan15], the two players, given their own input, try to guess the transcript  $\pi(x_1, x_2)$  of the protocol  $\pi$ . For this, player 1 picks a candidate  $t_1$  from the set  $\text{Im}(\pi(x_1, \cdot))$  of possible transcripts knowing that it has input  $x_1$ , while player 2 picks a candidate  $t_2$  from the set  $\text{Im}(\pi(\cdot, x_2))$ . The two players then communicate in order to find the first bit on which  $t_1$  and  $t_2$  disagree. The general structure of protocols ensures that the common prefix of  $t_1$  and  $t_2$  (until the first bit of disagreement) is identical to the beginning of the correct transcript on inputs  $x_1$  and  $x_2$ , i.e., identical to  $\pi(x_1, x_2)$ . Starting from this correct prefix, the players then pick new candidates for the transcript of the protocol  $\pi(x_1, x_2)$ , and so on, until they agree on the full transcript  $\pi(x_1, x_2)$ . Clever choices of the candidates, along with an efficient technique to find the first bit which differs between the candidates, lead to a protocol with little communication.

In extending the proof in [BBK<sup>+</sup>13, Pan15] to the multi-party case, new difficulties occur. The players can no longer try to guess the full transcript, as they have little information about the conversation between the other players, but can only try to guess their partial transcripts, according to their own input. Then, in order to find the first disagreement in the global transcript, every pair of players needs to find and communicate the place of the first disagreement in their partial transcripts. This induces the  $k^2$  factor in our compression scheme. It is unclear if this is necessary. Moreover, the players lack a common reference time. To solve this, we will introduce, as a technical tool in the proof, a *coordinator*, whose role is to introduce a round structure in the protocol  $\pi$ . Note that this is only in the proof, and that the stated results hold in the model we define in Section 3.

We will use a black box, call it *lcp box* (for *largest common prefix*), which can be used by two players  $A$  and  $B$  in the following way:  $A$  puts a string  $x$  in the box,  $B$  puts a string  $y$  in the box, and the box gives them back the first index  $j$  such that  $x_j \neq y_j$  if  $x \neq y$ , or tells them that  $x = y$  otherwise. The price to pay for using this black box is  $\log n$  bits of communication, where  $n = \max(|x|, |y|)$ .

This box can be efficiently simulated if we allow error:

**Lemma 6.5** ([FRPU94]). *There is a randomized public coin protocol such that on input two  $n$ -bits strings  $x$  and  $y$ , it outputs the first index  $j$  such that  $x_j = y_j$  with probability at least  $1 - \epsilon$  if such a  $j$  exists, and otherwise outputs that the two strings are equal, with worst case communication complexity  $\mathcal{O}(\log(n/\epsilon))$ .*

**Corollary 6.6** ([BBK<sup>+</sup>13]). *Any protocol  $\tilde{\rho}$  that uses an lcp box  $l$  times on average can be simulated with error  $\delta$  by a protocol  $\rho$  that does not use an lcp box, and communicates an average of  $\mathcal{O}(l \log(\frac{l}{\delta}))$  extra bits.*

We will use the lcp box in the proof, and use corollary 6.6 to perform the analysis at the end.

*Proof of theorem 6.4.* Starting from protocol  $\pi$ , we build a protocol  $\pi'$  which introduces a new player, called *coordinator*. In the beginning, the coordinator sends a signal message to each player. When a player  $i_1$  receives the signal from the coordinator, if he wants to send a bit  $m$  to some player  $i_2$  in  $\pi$ , in

$\pi'$  player  $i_1$  sends instead  $(m, i_2)$  to the coordinator (he can send messages to different players). If a player does not want to send any bit to the other players, he sends a “no” signal to the coordinator. When the coordinator has received an answer from all the players, he forwards the messages: if the coordinator has received some message  $(m, i_2)$  from  $i_1$ , he sends  $(m, i_1)$  to player  $i_2$ . If the coordinator has no message to send to a player, he sends him a “no” signal. This was the first round. Each player can now either send messages to be forwarded by the coordinator, either a “no” signal, and so on. Thus, in  $\pi'$  each player communicates only with the coordinator.  $\pi'$  computes the same function as  $\pi$  with the same error ratio, and  $\text{CC}(\pi') = \mathcal{O}(k\text{CC}(\pi))$ . Just as  $\pi$ ,  $\pi'$  uses only public randomness.

We can now define the protocol  $\tilde{\rho}$  (in which there is no coordinator). The players start by picking the shared randomness  $r$  as in protocol  $\pi'$ . For each  $i$ , define the set  $\mathcal{X}_i$  to be the set of possible inputs of player  $i$ , and the set  $\Pi_{(i)}(x_i)$  the set of possible transcripts of the communication between player  $i$  and the coordinator, knowing that player  $i$  has input  $x_i$ :

$$\Pi_{(i)}(x_i) = \overleftarrow{\pi'_i}(\mathcal{X}_1, \dots, \mathcal{X}_{i-1}, x_i, \mathcal{X}_{i+1}, \dots, \mathcal{X}_k, r).$$

We define a canonical way to write down the transcripts. A transcript  $t_i$  has the form  $t_i^1 \# t_i^2 \# \dots$  where  $t_i^m$  is a description of the partial transcript of player  $i$  of the  $m^{\text{th}}$  round of the protocol. We now define the partial transcripts. Formally, for  $i < j$ , denoting by  $t_i^m$  the partial transcript of player  $i$  of the transcript and by  $t_j^m$  the partial transcript of player  $j$  of the transcript within the  $m^{\text{th}}$  round, the messages sent by player  $i$  come before those sent by player  $j$  in  $t_i^m$  and  $t_j^m$ , and if  $j_1 < j_2$ , the messages sent to or received from player  $j_1$  come before the messages sent to or received from  $j_2$  in  $t_i^m$  for  $i \neq j_1, i \neq j_2$ .  $t_j^m$  is a sequence of messages of the form  $(b, i)$  where  $b$  is a bit and  $i$  is the index of the other player involved, and of  $\#$  which mark the separation between the messages exchanged with player  $i$  and the messages exchanged with player  $i+1$  and the separation between messages sent to player  $i$  and messages received from player  $i$ .

For example, the transcript of the  $m^{\text{th}}$  round by player 2 in a protocol with three players is of the form

$$t_2^m = (\text{messages by 1 to 2}) \# (\text{messages by 2 to 1}) \# (\text{messages by 2 to 3}) \# (\text{messages by 3 to 2}).$$

Note that for any  $t \in \Pi_{(i)}(x_i)$ , thanks to the  $\#$  messages, it is easy to associate to each message the round it has been sent or received.

Each player  $i$  represents  $\Pi_{(i)}(x_i)$  by a ternary tree  $T_i$  as follows.

1. The root is the largest common prefix (lcp) of the transcripts in  $\Pi_{(i)}(x_i)$ , and the remaining nodes are defined inductively.
2. If we have node  $\tau$ , then
  - the first child of  $\tau$  is the lcp of the transcripts in  $\Pi_{(i)}(x_i)$  beginning with  $\tau \circ (0, j)$ , *i.e.*  $\tau$  concatenated with the message  $(0, j)$ , where  $j$  is the next player to communicate with according to  $\tau$  (this  $j$  is determined by the number of  $\#$  in  $\tau$ ).

- the second child of  $\tau$  is the lcp of the transcripts in  $\Pi_{(i)}(x_i)$  beginning with  $\tau \circ (1, j)$ , where  $j$  is the next player to communicate with according to  $\tau$ .
  - the third child is  $\tau \circ \#$ .
3. The leaves are labelled by the possible transcripts of player  $i$ , *i.e.* the elements of  $\Pi_{(i)}(x_i)$ . We impose that each transcript ends with a “end” signal, thus a transcript cannot be the prefix of another transcript. The leaf  $f$  with label  $t_i$  has weight

$$w(t_i) = \Pr_{(X_j)_{j \neq i} | X_i = x_i} [\overleftarrow{\pi}_i'(X_1, \dots, X_{i-1}, x_i, X_{i+1}, \dots, X_k, r) = t_i].$$

The weight of a non-leaf node is defined by induction as the sum of the weights of its children. By construction, the weight of the root is 1.

We say that  $(t_1, \dots, t_k) \in \Pi_{(1)}(x_1) \times \dots \times \Pi_{(k)}(x_k)$  is a *coherent profile* if, for each round  $l$ , any message which appears in  $t_i$  as sent to player  $j$  also appears in  $t_j$  as coming from  $i$ . In fact,  $(\overleftarrow{\pi}_1'(x_1, \dots, x_k, r), \dots, \overleftarrow{\pi}_k'(x_1, \dots, x_k, r))$  is the only coherent profile. Indeed, take a coherent profile  $(t_1, \dots, t_k)$ . For each  $i$ ,  $t_i$  is of the form  $\overleftarrow{\pi}_i'(x_1^i, \dots, x_{i-1}^i, x_i, x_{i+1}^i, \dots, x_k^i)$  where  $\forall j \neq i, x_j^i \in X_j$ . Since whenever a player sends a message, his choice is based only on his own input and on the messages he has received before, it implies (by induction on the rounds of the protocol) that  $(t_1, \dots, t_k) = (\overleftarrow{\pi}_1'(x_1, \dots, x_k, r), \dots, \overleftarrow{\pi}_k'(x_1, \dots, x_k, r))$ .

We now show how the players can collaborate to find efficiently this coherent profile, that is how each player  $i$  can find  $\overleftarrow{\pi}_i'(x_1, \dots, x_k, r)$ . The players will proceed in stages  $s = 1, 2, \dots$

1. At the beginning of stage  $s$ , each player  $i$  is at a node  $\tau_i(s)$  such that  $(\tau_1(s), \dots, \tau_k(s))$  is a (term-wise) prefix of  $(\overleftarrow{\pi}_1'(x_1, \dots, x_k, r), \dots, \overleftarrow{\pi}_k'(x_1, \dots, x_k, r))$ . Player  $i$  starts the protocol with the node  $\tau_i(1)$ , which is the root of the tree  $T_i$ .
2. Each player  $i$  picks a candidate leaf  $t_i(s)$  in the tree  $T_i$ , such that  $t_i(s)$  is a successor of  $\tau_i(s)$ . We describe later how each player chooses his candidate leaf.
3. Each pair of players  $(i, j)$  will use an lcp box to find the first occurrence where  $t_i(s)$  is not coherent with  $t_j(s)$ , that is one of the players is supposed to send some message  $m$  in some round but the other is not supposed to receive it during this round, or one of the players is not supposed to send some message  $m$  in some round but the other is supposed to receive it during this round. If for all pairs of players there is no such occurrence, it means that  $(t_1(s), \dots, t_k(s))$  is a coherent profile, each player  $i$  has found  $\overleftarrow{\pi}_i'(x_1, \dots, x_k, r)$ , and the protocol terminates.
4. Denote by  $z_i$  the earliest round where player  $i$  disagrees with some other player. Then player  $i$  sends a signal to the first player (*i.e.* with the lowest index) he disagrees with during round  $z_i$ .



5. For any  $(i, j)$  such that each one has sent a signal to the other (note that there is at least one such pair  $(i, j)$ : each player receiving a signal also sends a signal, and the signals cannot form a cycle of length more than 2), players  $i$  and  $j$  exchange information about the first occurrence they found using the lcp box: each player tells the other what he wanted to send, or what he expected to receive. The player who has the sender role is always correct: for instance, if player  $i$  wanted to send 0 but player  $j$  expected to receive 1, or nothing (as encoded by  $\#$ ), then  $i$  is right, because what player  $i$  sent in the protocol  $\pi$  is based on the previous rounds (which are correct by hypothesis). Thus the player  $j$  having the receiver role is wrong. Player  $j$  updates his  $\tau_j$ : in  $T_j$ , starting from  $t_j(s)$ , he goes up toward  $\tau_j(s)$ , until he reaches a node  $\hat{\tau}_j$  which is correct (according to the result of the lcp box). Then, he defines  $\tau_j(s+1)$  as the correct child of  $\hat{\tau}_j$ , which he infers from the information exchanged with player  $i$ .
6. The players who have not updated their  $\tau_i$  yet define  $\tau_i(s+1) = \tau_i(s)$ .

First, note that the property of item (1) has been preserved for all players. This property ensures that the players will eventually agree on  $\overleftrightarrow{\pi}'_i(x_1, \dots, x_k, r)$ . We now specify how each player  $i$  chooses his candidate leaf at each stage. Player  $i$  picks the leaf  $t_i(s)$  which corresponds to the transcript with highest probability conditioned on the prefix specified by the node the player is at. Formally, at stage  $s$ , player  $i$  defines  $\tau^1 = \tau_i(s)$ , and then defines inductively  $\tau^{j+1}$  to be the child of  $\tau^j$  which has higher weight (breaking ties arbitrarily), until he reaches a leaf: this is the candidate  $t_i(s)$ .

Suppose that during round  $s$ , player  $i$  has participated in step (5) and has made the wrong choice (there must be at least one such player per round). We show that  $w(\tau_i(s+1)) \leq \frac{1}{2}w(\tau_i(s))$ . We look at the sequence  $(\tau^j)$  defined by player  $i$  when he chose his candidate leaf  $t_i(s+1)$  at step  $s$ . Let  $\tau^j$  be the first common ancestor of  $t_i$  and  $\tau_i(s+1)$ . By construction,  $\tau_i(s+1)$  is a direct child of  $\tau^j$ , and  $t_i$  is (a descendant of) another child of  $\tau^j$ . By the candidate leaf's construction process,  $w(\tau_i(s+1)) \leq \frac{1}{2}w(\tau^j) \leq \frac{1}{2}w(\tau_i(s))$ .

We conclude the analysis. On inputs  $(x_1, \dots, x_k)$ , let  $(t_1, \dots, t_k)$  denote the coherent profile. Each player will correct his  $\tau_i$  no more than  $\log \frac{1}{w(t_i)}$  times, because the weight of the node  $\tau_i$  halves with each correction, as noticed before, and given that the root has weight 1. Hence, the total number of corrections, and thus the number of stages  $S$ , is bounded by  $\sum_{i=1}^k \log \frac{1}{w(t_i)}$ . We take the average over inputs and shared randomness:

$$\begin{aligned}
\mathbb{E}_{r,x}[S] &= \mathbb{E}_{r,x} \sum_{i=1}^k \log \frac{1}{w(t_i)} \\
&= \sum_{i=1}^k \mathbb{E}_{r,x_i} \left[ \mathbb{E}_{(x_j)_{j \neq i} | X_i = x_i} \left[ \log \frac{1}{w(t_i)} \right] \right] \\
&= \sum_{i=1}^k \mathbb{E}_{r,x_i} \left[ \mathbb{E}_{(x_j)_{j \neq i} | X_i = x_i} \left[ \log \frac{1}{\Pr_{(X_j)_{j \neq i} | X_i = x_i} [\langle \overrightarrow{\Pi}_i' \rangle(X_1, \dots, X_{i-1}, x_i, X_{i+1}, \dots, X_k, r) = \langle \overrightarrow{\Pi}_i' \rangle(x_1, \dots, x_k, r)]} \right] \right] \\
&= \sum_{i=1}^k \mathbb{E}_{r,x_i} \left[ \mathbb{E}_{t_i | X_i = x_i, R=r} \left[ \log \frac{1}{\Pr_{(X_j)_{j \neq i} | X_i = x_i} [\langle \overrightarrow{\Pi}_i' \rangle(X_1, \dots, X_{i-1}, x_i, X_{i+1}, \dots, X_k, r) = t_i]} \right] \right] \\
&= \sum_{i=1}^k \mathbb{E}_{r,x_i} \left[ H(\langle \overrightarrow{\Pi}_i' \rangle | x_i r) \right] = \sum_{i=1}^k H(\langle \overrightarrow{\Pi}_i' \rangle | X_i R^p) = \sum_{i=1}^k I(X_{-i}; \langle \overrightarrow{\Pi}_i' \rangle | X_i R^p) = \text{IC}_\mu(\pi').
\end{aligned}$$

We have shown that the average number of stages is bounded by  $\text{IC}_\mu(\pi') \leq \text{IC}_\mu(\pi) \log(\text{CC}(\pi))$  as the only additional information players get in  $\pi'$  is the round of each message. At each stage, the communication consists of  $\frac{k(k-1)}{2}$  calls to the lcp box on strings of length at most  $\mathcal{O}(k\text{CC}(\pi))$  (one call for each pair of players), plus the  $k$  signals of step (4) and the exchange of information of step (5) (there are at most  $\frac{k}{2}$  pairs getting to step (5)). Hence

$$\begin{aligned}
\text{ACC}_\mu(\tilde{\rho}) &= \mathcal{O} \left( \text{IC}_\mu(\pi) \log(\text{CC}(\pi)) (k(k-1) \log(k\text{CC}(\pi)) + k + \frac{k}{2}) \right) \\
&= \mathcal{O} \left( k^2 \text{IC}_\mu(\pi) \log(k\text{CC}(\pi)) \log(\text{CC}(\pi)) \right).
\end{aligned}$$

Using corollary 6.6 we can replace each use of the lcp box with the protocol given by Lemma 6.6, to get the protocol  $\rho$  which simulates  $\pi$  with error  $\delta$  and average communication:

$$\begin{aligned}
\text{ACC}_\mu(\rho) &= \text{ACC}_\mu(\tilde{\rho}) + \mathcal{O} \left( k^2 \text{IC}_\mu(\pi) \log(\text{CC}(\pi)) \log \frac{k^2 \text{IC}_\mu(\pi) \log(\text{CC}(\pi))}{\delta} \right) \\
&= \mathcal{O} \left( k^2 \text{IC}_\mu(\pi) \log(\text{CC}(\pi)) \left( \log(k\text{CC}(\pi)) + \log \frac{k^2 \text{IC}_\mu(\pi) \log(\text{CC}(\pi))}{\delta} \right) \right).
\end{aligned}$$

□

## 6.2 A direct sum for PIC implies a direct sum for CC

**Theorem 6.7.** *Given a  $k$ -variable function  $f$  and a distribution  $\mu$  on inputs of  $f$ , if the existence of a protocol  $\pi$  computing  $f^{\otimes t}$  with error  $\epsilon \geq 0$  implies that there exists a protocol  $\pi'$  computing  $f$  with error  $\epsilon$  and verifying  $\text{PIC}_\mu(\pi') \leq \frac{1}{t} \text{PIC}_{\mu^{\otimes t}}(\pi)$ ,  $\text{CC}(\pi') \leq \text{CC}(\pi)$ , then*

$$\text{CC}^{2(\epsilon+\delta)}(f) = \mathcal{O} \left( \frac{k^2}{t(\epsilon+\delta)} \text{CC}^\epsilon(f^{\otimes t}) \log(\text{CC}^\epsilon(f^{\otimes t})) \log(k) \log \frac{k^2 \text{CC}^\epsilon(f^{\otimes t}) \log(\text{CC}^\epsilon(f^{\otimes t}))}{\delta} \right).$$

Note that the result of this theorem is meaningful when  $t$  is large with respect to  $k$ .

To prove this result, we first need the following fact.

**Fact 6.8.** Any 2-party protocol with error  $\frac{\epsilon}{2}$  and average communication complexity  $C$  can be turned into a protocol with distributional error  $\epsilon$  and worst case communication complexity  $\frac{2C}{\epsilon}$ .

We can generalize this fact to  $k$  party-protocols.

**Lemma 6.9.** Any  $k$ -party protocol with error  $\frac{\epsilon}{2}$  and average communication complexity  $C$  can be turned into a protocol with distributional error  $\epsilon$  and worst case communication complexity  $\frac{4C \log(k)}{\epsilon}$ .

*Proof.* Use one of the players as a relay. This increase the communication from  $C$  to  $2C \log(k)$ . We can then control the communication of the protocol just using fact 6.8. ┘

*Proof of theorem 6.7.* Let  $\mu$  be a distribution on inputs of  $f$ . Consider a protocol  $\pi$  computing  $f^{\otimes t}$  with error  $\epsilon$ . By hypothesis, there exist  $\pi'$  computing  $f$  with error  $\epsilon$  and verifying  $\text{PIC}_\mu(\pi') \leq \frac{1}{t} \text{PIC}_{\mu^{\otimes t}}(\pi)$ ,  $\text{CC}(\pi') \leq \text{CC}(\pi)$ . By theorem 4.6, we can impose  $\pi'$  to use only public randomness.

Applying successively theorem 6.7 and fact 6.8, we get a protocol  $\rho$  with distributional error  $2(\epsilon + \delta)$

$$\begin{aligned} \text{CC}(\rho) &= \mathcal{O} \left( \frac{1}{(\epsilon + \delta)} k^2 \text{PIC}_\mu(\pi') \log(\text{CC}(\pi')) \log(k) \left( \log(k \text{CC}(\pi')) + \log \frac{k^2 \text{PIC}_\mu(\pi') \log(\text{CC}(\pi'))}{\delta} \right) \right) \\ &= \mathcal{O} \left( \frac{1}{(\epsilon + \delta)} k^2 \text{PIC}_\mu(\pi') \log(\text{CC}(\pi)) \log(k) \log \frac{k^2 \text{CC}(\pi) \log(\text{CC}(\pi))}{\delta} \right). \end{aligned}$$

Thus

$$\begin{aligned} \text{CC}(\rho) &= \mathcal{O} \left( \frac{1}{t(\epsilon + \delta)} k^2 \text{PIC}_{\mu^{\otimes t}}(\pi) \log(\text{CC}(\pi)) \log(k) \log \frac{k^2 \text{CC}(\pi) \log(\text{CC}(\pi))}{\delta} \right) \\ &= \mathcal{O} \left( \frac{1}{t(\epsilon + \delta)} k^2 \text{CC}(\pi) \log(\text{CC}(\pi)) \log(k) \log \frac{k^2 \text{CC}(\pi) \log(\text{CC}(\pi))}{\delta} \right). \end{aligned}$$

As this reasoning is valid for any distribution  $\mu$ , Yao's minimax lemma implies

$$\text{CC}_{2(\epsilon + \delta)}(f) = \mathcal{O} \left( \frac{1}{t(\epsilon + \delta)} k^2 \text{CC}_\epsilon(f^{\otimes t}) \log(\text{CC}_\epsilon(f^{\otimes t})) \log(k) \log \frac{k^2 \text{CC}_\epsilon(f^{\otimes t}) \log(\text{CC}_\epsilon(f^{\otimes t}))}{\delta} \right)$$

as wanted. ┘

## 7 Tight lower bounds for the parity function Par

We now show how one can indeed use PIC to study multi-party communication protocols and to prove tight bounds. We study one of the canonical problems for zero-error multi-party computation, the parity function. The  $k$ -party parity problem with  $n$ -bit inputs  $\text{Par}_k^n$  is defined as follows. Each player  $i$  receives

$n$  bits  $(x_i^p)_{p \in [1, n]}$  and Player 1 has to output the bitwise XOR of the inputs  $\left( \bigoplus_{i=1}^k x_i^1, \bigoplus_{i=1}^k x_i^2, \dots, \bigoplus_{i=1}^k x_i^n \right)$ .

There is a simple private protocol for  $\text{Par}_k^n$  that uses  $n$  bits of private randomness. Player 1 uses a private random  $n$ -bit string  $r$  and sends to Player 2 the string  $x_1 \oplus r$ . Then, Player 2 computes the bit-wise parity of its input with the message and sends  $x_2 \oplus x_1 \oplus r$  to Player 3. The players continue until Player 1 receives back the message  $x_k \oplus \dots \oplus x_1 \oplus r$ . Player 1 then takes the bit-wise parity of this message with the private string  $r$  to compute the value of the parity function. It is easy to see that this protocol has information cost equal to  $n$ , since Player 1 just learns the value of the function and all other players learn nothing. We thus see that information cost cannot provide here lower bounds that scale with  $k$ .

We now prove tight lower bounds for this problem using the measure of PIC. For this, we study a restricted class of protocols: we only consider protocols such that for any player  $i$ , the sets  $(S_i^l)_l$  and  $(\overleftarrow{S}_i^l)_l$  do not depend on the input  $x$  or on the randomness. In other words, the structure of the protocol is fixed and independent of the input and randomness. Note that the private protocol we described above fits in this model.

Our bound for  $\text{Par}_k^n$  is in fact proved for a wider class of protocols, where we allow the player outputting  $\bigoplus_{i=1}^k x_i^p$  to be different for each coordinate  $p$  and to depend on the input.

**Theorem 7.1.**  $\text{PIC}_\mu^0(\text{Par}_k^n) \geq n(k-1)$  where  $\mu$  is the uniform input distribution.

*Proof.* We use the uniform distribution  $\mu$  throughout the proof. Since we are looking at zero-error protocols, the public information cost is equal to the information cost of deterministic protocols. Let  $\pi$  be a protocol solving  $\text{Par}_k^n$  for  $k$  players and  $n$  bits per player. For any  $i$ , we define  $\overleftrightarrow{\Pi}_i$  to be the transcript of player  $i$ , i.e. all messages he sends and receives. We split this transcript into the messages he sends and the messages he receives. Denote by  $(M_i^{\overrightarrow{l}})_{l \geq 0}$  the set of messages sent by player  $i$  in the protocol  $\pi$ , ordered by round and inside each round ordered by index of the recipient. Similarly, denote by  $(M_i^{\overleftarrow{l}})_{l \geq 0}$  the set of messages received by player  $i$ . Denote by  $j(i, l)$  the player receiving  $M_i^{\overrightarrow{l}}$ .

For any  $l_0$ , let  $M_i^{\overrightarrow{l_0}}$  be the random variable representing the so-far history, i.e. all messages to and from player  $i$  until the time of message  $M_i^{\overrightarrow{l_0}}$ . In a similar way, define  $M_i^{\overleftarrow{l_0}}$  to be the random variable representing the history of the messages to and from player  $i$  until the time of message  $M_i^{\overleftarrow{l_0}}$ . We also define a function  $l'(i, l)$  such that every message  $M_i^{\overrightarrow{l}}$  sent by player  $i$  is received by player  $j(i, l)$  as  $M_j^{\overleftarrow{l'}}$  where  $l'$  is a shortcut for  $l'(i, l)$  and  $j$  for  $j(i, l)$ .

Our first goal is to prove that

$$\text{PIC}_\mu^0(\pi) \geq \sum_i I(X_i; \overleftrightarrow{\Pi}_i).$$

Intuitively, this means that PIC at least takes into account the information each player leaks about his input to someone who has access to all messages that involve this player.

Note that we have for all  $i$

$$I(X_i; \overleftarrow{\Pi}_i) = \sum_l I(X_i; M_i^{\vec{l}} | M_i^{\leftarrow l}) + \sum_l I(X_i; M_i^{\leftarrow l} | M_i^{\vec{l}}) = \sum_l I(X_i; M_i^{\vec{l}} | M_i^{\leftarrow l}),$$

where we used the fact that every term of the second sum is 0, as for any  $l$ , conditioned on  $M_i^{\leftarrow l}$ ,  $X_i$  is independent of the variable  $M_i^{\vec{l}}$ .

Starting from the definition of PIC and using the chain rule, we can decompose it as a sum over all messages received in the protocol:

$$\text{PIC}_\mu^0(\pi) = \sum_i \sum_l I(X_{-i}; M_i^{\leftarrow l} | M_i^{\vec{l}} X_i).$$

We rearrange the sum by considering the messages from the point of view of the sender rather than the receiver.

$$\text{PIC}_\mu^0(\pi) = \sum_i \sum_l I(X_{-j}; M_i^{\vec{l}} | M_j^{\leftarrow l} X_j).$$

We will show that for any message  $M_i^{\vec{l}}$ ,  $I(X_{-j}; M_i^{\vec{l}} | M_j^{\leftarrow l} X_j) \geq I(X_i; M_i^{\vec{l}} | M_i^{\leftarrow l})$ .

As  $M_i^{\vec{l}}$  is determined by  $X_i$  and  $M_i^{\leftarrow l}$ ,  $H(M_i^{\vec{l}} | X_i M_i^{\leftarrow l}) = 0$ , and we have  $I(X_i; M_i^{\vec{l}} | M_i^{\leftarrow l}) = H(M_i^{\vec{l}} | M_i^{\leftarrow l})$ , and similarly  $I(X_{-j}; M_i^{\vec{l}} | M_j^{\leftarrow l} X_j) = H(M_i^{\vec{l}} | M_j^{\leftarrow l} X_j)$ . Thus

$$\begin{aligned} I(X_i; M_i^{\vec{l}} | M_i^{\leftarrow l}) \leq I(X_{-j}; M_i^{\vec{l}} | M_j^{\leftarrow l} X_j) &\iff H(M_i^{\vec{l}} | M_i^{\leftarrow l}) \leq H(M_i^{\vec{l}} | M_j^{\leftarrow l} X_j) \\ &\iff I(M_i^{\vec{l}}; M_i^{\leftarrow l}) \geq I(M_i^{\vec{l}}; M_j^{\leftarrow l} X_j). \end{aligned}$$

We show that  $I(M_i^{\vec{l}}; M_i^{\leftarrow l}) = I(M_i^{\vec{l}}; M_i^{\leftarrow l} M_j^{\leftarrow l} X_j)$ , which implies that the last inequality is true. For this we just need to check that  $I(M_i^{\vec{l}}; M_j^{\leftarrow l} X_j | M_i^{\leftarrow l}) = 0$ . To see this, notice that given the value of  $M_i^{\leftarrow l}$ ,  $M_i^{\vec{l}}$  is determined by  $X_i$  and thus  $I(X_i; M_j^{\leftarrow l} X_j | M_i^{\leftarrow l}) \geq I(M_i^{\vec{l}}; M_j^{\leftarrow l} X_j | M_i^{\leftarrow l})$ , and notice that  $I(X_i; M_j^{\leftarrow l} X_j | M_i^{\leftarrow l}) = 0$  (cf. lemma A.2 in the appendix).

We have shown so far that  $\text{PIC}_\mu^0(\pi) \geq \sum_i I(X_i; \overleftarrow{\Pi}_i)$ . Showing that  $I(X_i; \overleftarrow{\Pi}_i) \geq n(k-1)$  will conclude the proof.

Let  $x = (x_i^p) \in \{0,1\}^{nk}$ , where  $x_i$  is the  $n$ -bit input of player  $i$ . For any  $p$ , let  $q^p(x)$  be the index of the first player able to compute  $\bigoplus_{i=1}^k x_i^p$  (formally,

we say that a player is able to compute  $\bigoplus_{i=1}^k x_i^p$  at some time if the value of

$\bigoplus_{i=1}^k x_i^p$  is fixed given his input and his transcript until that time). For any  $i$ , define  $C_i(x) = \{p, q^p(x) \neq i\}$ , which represents the coordinates of his input that player  $i$  is intuitively going to leak when the players are given input  $x$ , and let  $c_i(x) = |C_i(x)|$ .

We show that  $\forall i, H(X_i \mid \overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)) \leq n - c_i(x)$ . Assume towards a contradiction that for some  $i$ ,  $H(X_i \mid \overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)) > n - c_i(x)$ . This implies that the number of possible values for  $X_i$  consistent with  $\overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)$  is more than  $2^{n-c_i(x)}$ , and thus the number of coordinates of the input of the  $i$ -th player that are fixed by the transcript is strictly less than  $c_i(x)$ . Then there must exist  $x'$  such that

- $\overleftrightarrow{\pi}_i(x') = \overleftrightarrow{\pi}_i(x)$
- $\exists p \in C_i(x)$  such that  $x_i^p \neq x_i^p$ .

Note that  $\overleftrightarrow{\pi}_i(x') = \overleftrightarrow{\pi}_i(x)$  implies (as a general property of communication protocols) that  $\overleftrightarrow{\pi}_i(x) = \overleftrightarrow{\pi}_i(x'_i, x_{-i})$ . As  $q^p(x) \neq i$ , this is a contradiction, since then player  $q^p(x)$  would output the same answer on  $x$  and  $(x'_i, x_{-i})$ , while  $\bigoplus_{j=1}^k x_j^p \neq x_i^p \oplus \bigoplus_{j \neq i} x_j^p$ .

Thus  $\forall i, H(X_i \mid \overleftrightarrow{\Pi}_i = \overleftrightarrow{\pi}_i(x)) \leq n - c_i(x)$  and  $H(X_i \mid \overleftrightarrow{\Pi}_i) \leq \mathbb{E}_x[n - c_i(x)] = n - \mathbb{E}_x[c_i(x)]$ . Thus  $I(X_i; \overleftrightarrow{\Pi}_i) \geq \mathbb{E}_x[c_i(x)]$ .

Summing over all  $i$ , we get  $\sum_i I(X_i; \overleftrightarrow{\Pi}_i) \geq \sum_i \mathbb{E}_x[c_i(x)] = \mathbb{E}_x[\sum_i c_i(x)]$  and since  $\sum_i c_i(x) = n(k-1)$  for any  $x$ , we get  $\sum_i I(X_i; \overleftrightarrow{\Pi}_i) \geq \mathbb{E}_x[n(k-1)] = n(k-1)$ .

As  $\text{PIC}_\mu^0(\pi) \geq \sum_i I(X_i; \overleftrightarrow{\Pi}_i)$ , we have shown that  $\text{PIC}_\mu(\pi) \geq n(k-1)$ .  $\lrcorner$

We can now lower bound the amount of randomness in any private computation of  $\text{Par}_k^n$ .

**Theorem 7.2.** *The entropy in the private randomness of a private protocol for  $\text{Par}_k^n$  is at least  $\frac{k-2}{k-1}n$ .*

*Proof.* For  $\text{Par}_k^n$ , where one player outputs the parity for each coordinate, we have  $\sum_i H(f_i) = n$ . Applying corollary 5.2, we get:  $H(\Pi \mid XR^p) \geq n \frac{k-2}{k-1}$ .  $\lrcorner$

Using Theorem 5.1 we can also give a lower bound on the randomness one needs for protocols that are allowed to leak some limited amount of information about the inputs of the players.

## 8 Conclusions

In this paper we introduce a new information-theoretic measure, that we call PIC, for the study of multi-party computation protocols in the number-in-hand, peer-to-peer model. This is probably the most natural (distributed) computation model, and also closely related to the models used in the distributed algorithms community. Previous information-theoretic measures that were used successfully for the study of two-party computation protocols do not extend immediately to the multi-party case due to the fact that private protocols exist for any function in the multi-party setting [BOGW88, CCD88]. Our notion of PIC

provides an alternative way of studying multi-party protocols, especially when one is interested in notions of privacy. Furthermore, PIC may yield tight results for certain functions, for which using other models, such as the coordinator model, would imply a loss of a logarithmic factor.

We define this measure in a slightly restricted computation model which however still allows private protocols, and applies to almost any protocol in the literature. We prove a number of properties of our new measure, PIC, and a number of connections to other complexity notions, e.g., the amount of randomness needed for private computation or the central question of direct sum (in communication complexity).

Our work opens the way to interesting directions for further work. A challenging direction would be to prove a tight lower bound for Disjointness in the message passing peer-to-peer model (without the loss of a logarithmic factor). An even more ambitious goal would be to use our result from Section 6 to try and prove the direct sum property for communication complexity, in either the two-party or multi-party setting, via our measure of PIC.

## References

- [BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. How to compress interactive communication. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 67–76, New York, NY, USA, 2010. ACM.
- [BBK<sup>+</sup>13] Joshua Brody, Harry Buhrman, Michal Koucký, Bruno Loff, Florian Speelman, and Nikolay K. Vereshchagin. Towards a reverse newman’s theorem in interactive information complexity. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 24–33. IEEE, 2013.
- [BCKO93] Reuven Bar-Yehuda, Benny Chor, Eyal Kushilevitz, and Alon Or-litsky. Privacy, additional information and communication. *IEEE Transactions on Information Theory*, 39(6):1930–1943, 1993.
- [BEO<sup>+</sup>13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 668–677. IEEE Computer Society, 2013.
- [BG14] Mark Braverman and Ankit Garg. Public vs private coin in bounded-round information. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 502–513. Springer, 2014.



- [BGPW13] Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, STOC '13, pages 151–160, New York, NY, USA, 2013. ACM.
- [BMY15] Balthazar Bauer, Shay Moran, and Amir Yehudayoff. Internal compression of protocols to entropy. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPICs*, pages 481–496. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- [BO15] Mark Braverman and Rotem Oshman. On information complexity in the broadcast model. In Chryssis Georgiou and Paul G. Spirakis, editors, *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 355–364. ACM, 2015.
- [BOGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 1–10, New York, NY, USA, 1988. ACM.
- [BR11] Mark Braverman and Anup Rao. Information equals amortized communication. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS '11*, pages 748–757, Washington, DC, USA, 2011. IEEE Computer Society.
- [Bra12] Mark Braverman. Interactive information complexity. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 505–524, New York, NY, USA, 2012. ACM.
- [BYJS02] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. In *Proceedings of the 43rd Symposium on Foundations of Computer Science, FOCS '02*, pages 209–218, Washington, DC, USA, 2002. IEEE Computer Society.
- [CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 11–19, New York, NY, USA, 1988. ACM.
- [CKS03] Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *In IEEE Conference on Computational Complexity*, pages 107–117, 2003.

- [CM15] Arkadev Chattopadhyay and Sagnik Mukhopadhyay. Tribes is hard in the message passing model. In Ernst W. Mayr and Nicolas Ollinger, editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, volume 30 of *LIPIcs*, pages 224–237. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- [CR15] Arkadev Chattopadhyay and Atri Rudra. The range of topological effects on communication. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 540–551. Springer, 2015.
- [CRR14] Arkadev Chattopadhyay, Jaikumar Radhakrishnan, and Atri Rudra. Topology matters in communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 631–640, 2014.
- [CSWY01] Amit Chakrabarti, Yaoyun Shi, Anthony Wirth, and Andrew Chi-Chih Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *FOCS*, pages 270–278, 2001.
- [DF89] Danny Dolev and Tomás Feder. Multiparty communication complexity. In *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989*, pages 428–433. IEEE Computer Society, 1989.
- [FHW12] Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1150–1162. SIAM, 2012.
- [FJK<sup>+</sup>15] Lila Fontes, Rahul Jain, Iordanis Kerenidis, Sophie Laplante, Mathieu Laurière, and Jérémie Roland. Relative discrepancy does not separate information and communication complexity. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 506–516. Springer, 2015.
- [FKN94] Uri Feige, Joe Killian, and Moni Naor. A minimal model for secure computation (extended abstract). In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing, STOC '94*, pages 554–563, New York, NY, USA, 1994. ACM.
- [FKNN95] Tomás Feder, Eyal Kushilevitz, Moni Naor, and Noam Nisan. Amortized communication complexity. *SIAM J. Comput.*, 24(4):736–750, 1995.

- [FRPU94] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, October 1994.
- [GG10] Anna Gál and Parikshit Gopalan. Lower bounds on streaming algorithms for approximating the length of the longest increasing subsequence. *SIAM J. Comput.*, 39(8):3463–3479, 2010.
- [GKR14] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 176–185, 2014.
- [GKR15a] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of communication and external information. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:88, 2015.
- [GKR15b] Anat Ganor, Gillat Kol, and Ran Raz. Exponential separation of information and communication for boolean functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 557–566. ACM, 2015.
- [Gro09] Andre Gronemeier. Asymptotically optimal lower bounds on the  $\text{nih}$ -multi-party information complexity of the and-function and disjointness. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, volume 3 of *LIPICs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.
- [HJMR10] Prahladh Harsha, Rahul Jain, David A. McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. *IEEE Transactions on Information Theory*, 56(1):438–449, 2010.
- [Jai15] Rahul Jain. New strong direct product results in communication complexity. *J. ACM*, 62(3):20, 2015.
- [Jay09] T. S. Jayram. Hellinger strikes back: A note on the multi-party information complexity of and. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX '09 / RANDOM '09, pages 562–573, Berlin, Heidelberg, 2009. Springer-Verlag.
- [JRS03] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. A direct sum theorem in communication complexity via message compression. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 300–315. Springer, 2003.

- [Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 77–86. ACM, 2010.
- [KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.
- [Koz15] Alexander Kozachinskiy. *Computer Science – Theory and Applications: 10th International Computer Science Symposium in Russia, CSR 2015, Listvyanka, Russia, July 13-17, 2015, Proceedings*, chapter Making Randomness Public in Unbounded-Round Information Complexity, pages 296–309. Springer International Publishing, Cham, 2015.
- [MNSW95] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing, STOC ’95*, pages 103–111, New York, NY, USA, 1995. ACM.
- [Pan15] Denis Pankratov. *Communication complexity and information complexity*. PhD thesis, The university of Chicago, 2015.
- [PVZ12] Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’12*, pages 486–501. SIAM, 2012.
- [RS15] Anup Rao and Makrand Sinha. Simplified separation of information and communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:57, 2015.
- [Sha48] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.
- [Sha03] Ronen Shaltiel. Towards proving strong direct product theorems. *Computational Complexity*, 12(1-2):1–22, 2003.
- [SHK<sup>+</sup>10] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *CoRR*, abs/1011.3049, 2010.
- [WZ14] David P. Woodruff and Qin Zhang. An optimal lower bound for distinct elements in the message passing model. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 718–733. SIAM, 2014.
- [Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing, STOC ’79*, pages 209–213, New York, NY, USA, 1979. ACM.

## A Technical statements

**Lemma A.1.** *In the conditions of the proof of theorem 5.1,*

$$I(R_{-i}; R_i \mid X\Pi R^p) = 0.$$

*Proof.*

$$\begin{aligned} I(R_i; R_{-i} \mid X\Pi R^p) &= H(R_i \mid X\Pi R^p) - H(R_i \mid R_{-i}X\Pi R^p) \\ &= -H(R_i \mid X R^p) + H(R_i \mid X\Pi R^p) + H(R_i \mid X R^p R_{-i}) - H(R_i \mid R_{-i}X\Pi R^p) \\ &\quad (\text{As } R_i, R_{-i}, R^p \text{ and } X \text{ are independent}) \\ &= -I(R_i; \Pi \mid X R^p) + I(R_i; \Pi \mid X R^p R_{-i}). \end{aligned}$$

Thus we just need to prove that  $I(R_i; \Pi \mid X R^p R_{-i}) \leq I(R_i; \Pi \mid X R^p)$ . From now on the proof is similar to the proof that the internal IC of a protocol is lower than its external IC (cf. [Bra12]).

Let us write  $\Pi = \Pi^1 \dots \Pi^t$  where the  $\pi^p$  represent the messages ordered by round, and let  $\Pi^{<p} = \Pi^1 \dots \Pi^{p-1}$ .

Using the chain rule,  $I(R_i; \Pi \mid X R^p R_{-i}) = \sum_{p=1}^t I(R_i; \Pi^p \mid X R^p R_{-i} \Pi^{<p})$  and

$$I(R_i; \Pi \mid X R^p) = \sum_{p=1}^t I(R_i; \Pi^p \mid X R^p \Pi^{<p}).$$

We prove the inequality

$$\sum_{p=1}^t I(R_i; \Pi^p \mid X R^p R_{-i} \Pi^{<p}) \leq \sum_{p=1}^t I(R_i; \Pi^p \mid X R^p \Pi^{<p})$$

term-wise, for any  $p$ . If the sender of  $\Pi^p$  is not player  $i$ , as  $\Pi^p$  is a function of  $(X, R^p, R_{-i}, \Pi_i^{<p})$ ,  $I(R_i; \Pi_i^p \mid X R^p R_{-i} \Pi_i^{<p}) = 0$  and the inequality holds. Similarly, if  $\Pi_i^p$  is sent by player  $i$ ,  $I(R_{-i}; \Pi_i^p \mid X R^p R_{-i} \Pi_i^{<p}) = 0$  and applying lemma 2.8, we get that  $I(R_i; \Pi_i^p \mid X R^p R_{-i} \Pi_i^{<p}) \leq I(R_i; \Pi_i^p \mid X R^p \Pi_i^{<p})$ .  $\perp$

**Lemma A.2.** *In the conditions of the proof of theorem 7.1,*

$$I(X_i; M_j^{\leftarrow \vec{l}} X_j \mid M_i^{\leftarrow \vec{l}}) = 0.$$

*Proof.* Note that  $(M_j^{\leftarrow \vec{l}}, X_j)$  is a function of  $(X_{-i}, M_i^{\leftarrow \vec{l}})$ . The data processing inequality 2.7 implies that  $I(X_i; M_j^{\leftarrow \vec{l}} X_j \mid M_i^{\leftarrow \vec{l}}) \leq I(X_i; X_{-i} M_i^{\leftarrow \vec{l}} \mid M_i^{\leftarrow \vec{l}})$  and thus  $I(X_i; M_j^{\leftarrow \vec{l}} X_j \mid M_i^{\leftarrow \vec{l}}) \leq I(X_i; X_{-i} \mid M_i^{\leftarrow \vec{l}})$ .

We show that  $I(X_i; X_{-i} \mid M_i^{\leftarrow \vec{l}}) = 0$ , which will conclude the proof. The situation is very similar to the one of lemma A.1.

$$\begin{aligned} I(X_i; X_{-i} \mid M_i^{\leftarrow \vec{l}}) &= H(X_i \mid M_i^{\leftarrow \vec{l}}) - H(X_i \mid X_{-i} M_i^{\leftarrow \vec{l}}) \\ &= -H(X_i) + H(X_i \mid M_i^{\leftarrow \vec{l}}) + H(X_i \mid X_{-i}) - H(X_i \mid X_{-i} M_i^{\leftarrow \vec{l}}) \quad (X_i, X_{-i} \text{ independent}) \\ &= -I(X_i; M_i^{\leftarrow \vec{l}}) + I(X_i; M_i^{\leftarrow \vec{l}} \mid X_{-i}) \end{aligned}$$

Thus we just need to prove that  $I(X_i; M_i^{\leq \vec{l}} | X_{-i}) \leq I(X_i; M_i^{\leq \vec{l}})$ . From now on the proof is similar to the proof that the internal IC of a protocol is lower than its external IC (cf. [Bra12]).

Let us write  $M_i^{\leq \vec{l}} = M_i^1 \dots M_i^t$ , and let  $M_i^{< p} = M_i^1 \dots M_i^{p-1}$ . Using the chain rule,

$$I(X_i; M_i^{\leq \vec{l}} | X_{-i}) \leq I(X_i; M_i^{\leq \vec{l}}) \iff \sum_{p=1}^t I(X_i; M_i^p | X_{-i} M_i^{< p}) \leq \sum_{p=1}^t I(X_i; M_i^p | M_i^{< p})$$

We prove the inequality term-wise, for any  $p$ . If  $M_i^p$  is received by player  $i$ , as  $M_i^p$  is a function of  $(X_{-i}, M_i^{< p})$ ,  $I(X_i; M_i^p | X_{-i} M_i^{< p}) = 0$  and the inequality holds. Similarly, if  $M_i^p$  is sent by player  $i$ ,  $I(X_{-i}; M_i^p | X_i M_i^{< p}) = 0$  and applying lemma 2.8, we get that  $I(X_i; M_i^p | X_{-i} M_i^{< p}) \leq I(X_i; M_i^p | M_i^{< p})$ . ┘